286079

JPRS 84064

8 August 1983

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

No. 73

FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

PROCUREMENT OF PUBLICATIONS

JPRS publications may be ordered from the National Technical Information Service (NTIS), Springfield, Virginia 22161. In ordering, it is recommended that the JPRS number, title, date and author, if applicable, of publication be cited.

Current JPRS publications are announced in Government Reports Announcements issued semimonthly by the NTIS, and are listed in the Monthly Catalog of U.S. Government Publications issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Correspondence pertaining to matters other than procurement may be addressed to Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201.

JPRS REPORTS

Japan Report
Korean Affairs Report
Southeast Asia Report
Mongolia Report

Near East/South Asia Report
Sub-Saharan Africa Report
West Europe Report
West Europe Report: Science and Technology

Latin America Report

USSR

Political and Sociological Affairs
Problems of the Far East
Science and Technology Policy
Sociological Studies
Translations from KOMMUNIST
USA: Economics, Politics, Ideology
World Economy and International Relations
Agriculture
Construction and Related Industries
Consumer Goods and Domestic Trade
Economic Affairs
Energy
Human Resources
International Economic Relations
Transportation

Physics and Mathmetics
Space
Space Biology and Aerospace Medicine
Military Affairs
Chemistry
Cybernetics, Computers and Automation Technology
Earth Sciences
Electronics and Electrical Engineering
Engineering and Equipment
Machine Tools and Metal-Working Equipment
Life Sciences: Biomedical and Behavioral Sciences
Life Sciences: Effects of Nonionizing Electromagnetic Radiation
Materials Science and Metallurgy
Meteorology and Hydrology

EASTERN EUROPE

Political, Sociological and Military Affairs
Scientific Affairs

Economic and Industrial Affairs

CHINA

Political, Sociological and Military Affairs
Economic Affairs
Science and Technology

RED FLAG
Agriculture
Plant and Installation Data

WORLDWIDE

Telecommunications Policy, Research and Development
Nuclear Development and Proliferation

Epidemiology

FBIS DAILY REPORT

China
Soviet Union
South Asia
Asia and Pacific

Eastern Europe
Western Europe
Latin America
Middle East and Africa

To order, see inside front cover

JPRS 84064

8 August 1983

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

No. 73

CONTENTS

[III - USSR - 21C S&T]

CONFERENCES AND EXHIBITIONS

PUBLICATIONS

GENERAL

GEORGIA'S PROBLEMS WITH ORGANIZATION OF COMPUTER ACTIVITIES

Tbilisi ZARYA VOSTOKA in Russian 19 Apr 83 p 2

[Article by Temur Ratiani, deputy chief of the TsSU of the GeSSR, chief of the Main Administration of Computational Projects and Meri Broladze, chief of the Department of Computational Projects and ASU in national economy of the Main Administration of Computational Projects: "Achievements in Science and Technology - in all Economic Branches; Increasing Capabilities of Computers"]

[Text] One of the current means for improving the economic mechanism is the development of machine technology for processing and utilizing management information. The role of computer centers (VTs) and the significance of problems associated with their efficient utilization increases under these circumstances. Computer centers, which serve as the technical foundation for Automated Control Systems (ASU), are industrial subdivisions with different technological equipment, structure and purposes. Some of them are self-supporting, filling orders from customers; others are incorporated into an organizational structure of enterprises, industrial associations or economic branches. It is therefore no accident that they are being created at a rapidly accelerating pace.

The overall fleet of computers in our republic has increased several-fold during the last 10 years.

As the applications of computers change, so do the aims, forms and methods of their use. Although, in the past, only an insignificant number of computers were used in ASU, at present, more than one quarter of the total fleet is working within the system.

In recent years, the republic has witnessed the replacement of low-powered computers by modern ones belonging to the third generation. Unified systems of Electronic Computers (YeS EVM) of the "Ryad" type. Computers of this type encompass a wide range of high-speed machines. One of the main advantages of computers of the "Ryad" type is their program compatability which permits the development of programs independently

of the specific type of computer and its configuration, the utilization of general operating systems and the creation of a unified bank of packages of applied programs. This allows the consolidation of computer centers of all ASU into a single state system.

Computers are utilized most efficiently in industry, transportation and management. In computer centers of the Administration of the Transcaucasian Railway, the Central Statistical Administration, Ministry of Motor Transport of the Republic, Rustav Metallurgical Factory and Ministry of Construction of the Georgian SSR, a number of rather complex national-economic problems are being solved and a definite economic effect has been achieved.

At the same time, serious deficiencies as far as efficient utilization of computers is concerned, do exist, which we would like to discuss in some detail. In 1977, in view of a resolution of the Council of Ministers of the USSR, attention was focused on the necessity of increasing, in every possible manner, the efficiency of use of computer technology in the national economy. This required ministries and agencies to carry out a number of specific measures. An analysis of computer use in the republic shows that many ministries and agencies have not drawn the appropriate conclusions from these directives. The majority of computers are not being utilized to their fullest capacity. Thus the computers of the "Ryad" type work only from seven to ten hours per twenty-four hour period. The working time-table is often not optimal. For these very reasons, inspite of a low load, many enterprises and organizations are compelled to purchase additional, very expensive computers in order to handle the information during peak hours.

The topics of problems solved in the majority of enterprises encompassed in ASU are quite narrow; only accounting-statistical problems are being tackled using computers. The level of qualifications of the technical staff is inadequate. Out of 7,000 experts working at computer centers of the national economy of the Georgian SSR, only 58 percent have a higher education.

At present, computer performance is evaluated, taking into account its average load per twenty-four hour period. This index does not really characterize accurately the degree of their utilization. One cannot ignore the factor that different programs require different amounts of time to solve the very same problem. Moreover, the bulk of the working time is spent debugging and compiling programs and only an insignificant portion is spent on the actual problem solving; also the input and output procedures consume a substantial amount of time. The proportion of time spent in 1982 on (actual) solution of problems is only slightly more than one half of the total working time. This means that inspite of the high load, the technological capabilities of computers are not being utilized to their full extent. It would be desirable to develop an index which would take into account not only the working time of a computer but also the volume of computing performed.

The tendency of ministries and agencies of the republic to establish computer centers and to acquire their own computers without taking necessary preliminary measures to assure their prompt utilization resulted in inefficient use of

computers, in large amounts of unutilized machine time in the republic, in idleness of expensive computers and a deficit of highly qualified personnel. The current situation is that every enterprise, NII [Scientific-Research Institute], KB [Planning and Design Office] organize their own computer center which is often not staffed with appropriate personnel and lacks the facilities for its proper operation. As a result, many low-capacity computer centers are being established in which expensive computer technology remains idle for long periods of time and even after being put into service, the computers increase their pace very slowly. Although in comparison with last year, the idle time of computers has decreased, it is still quite substantial. The largest amount of idle time exists at the Computer and Information Center (IVTs) of the Georgian Maritime Steamship Line, the Republic Computer and Information Center (RIVTs) of the Ministry of Trade, the Republic Computer Center of the Ministry of Agriculture and the NII [Scientific Research Institute] "Mion". Based on available calculations, just for "Ryad" type computers, the amount of projects carried out could be increased by 6.4 million rubles or by 22.2 percent, provided a normative load could be maintained. The load of the computer at the initial stages of collective use become an important factor for the solution of this problem. In this case, a user does not purchase a computer but leases machine time from other computer centers and thus has no additional operational expenses.

Almost every ministry in our republic, in order to form an economic branch ASU tries to acquire not only its own computer center but also its own "supportive bases" in the outlying districts, which are called branches, groups, divisions, communication points, etc. For example, the Ministry of Motor Transport of the Republic creates its own network of computing organizations on the regional level. Motor transport enterprises were previously served by the computer system of the TsSU [Central Statistical Administration]. The processing of information at the Motor Transport Enterprises of Mtskhet and Tsnori was originally carried out by 6-8 operators at a cost of 500-600 rubles a month. Now, however, in the "groups" established by the Ministry of Motor Transport, the number of operators has increased by 20-30 persons and the expenses have risen to 4,000 rubles. And how many such "bases" does this Ministry own in the whole republic? The available data indicates that close to 400 people are engaged in this operation.

As early as 1977 a resolution was passed by the higher government bodies of the Georgian SSR restricting the establishment of agency installations on a regional level and instructing ministries and agencies to utilize widely the network of territorial computer installations of the TsSU [Central Statistical Administration]. In 1979, the governing board of Tsekavshiri opened a computer center in the city of Kutaisi, consisting of 96 units to process economic information for the Kutaisi, Samtred and Zugdid interdistrict wholesale bases without assuring appropriate facilities for their operation (space, personnel). The construction of a building for the computer center started in 1977, has dragged on and has not been completed yet. The computer YES-1022 which was purchased in the third quarter of 1980 and additional equipment costing a total of 800,000 rubles became operational only at the beginning of 1983. Consequently, the computer

was idle for three years and it is still unknown when it will be working at full capacity. Such occurrences, unfortunately, are not is lated. In our opinion, it would be desirable to entrust the regional section of the computer system of the TsSU [Central Statistical Administration] with the task of processing information of enterprises and organizations in the outlying districts as well as its transmission to the higher level branch ASU especially since the complex program ASU "Gruzia" stipulates that the system of TsSU should become the technological basis on the regional level. The computer system of the TsSU of the republic encompasses practically all the administrative districts and cities of the republic and functions on the basis of cost accounting. Hundreds of organizations arebbeing served by this system.

Along with the district section, it is necessary to concentrate forces on the republic level as well, since the computing capacity is also being utilized inappropriately here. Ministries and agencies devote too much attention to the establishment of new computer centers and the growth in the fleet of computers without taking appropriate steps to increase the efficiency of those already in service, resulting in an unnecessary expenditure of government resources.

Collective-use computer centers (VTsKP's) should become the most expedient form of computer usage. Such centers will help to solve the troublesome problem of space, handle properly an uninterrupted supply of spare parts and peripheral equipment. Finally it would also be easier to solve the personnel problem, which is quite acute due to the rapid change of generations of computers. In accordance with the resolution of the Council of Ministries of the USSR, the establishment of computer centers of national economy of the republic is being carried out in coordination with the TsSU, Gosplan and the Ministry of Finances of the USSR, while VTsKP's are being coordinated with the State Committee of the USSR for Science and Technology. There is no doublt that further improvement in the use of computers and ASU with simultaneous elimination of the shortcomings indicated above will allow us to improve the system of planning and management and will contribute to the further development of the socialist economy.

9966
CSO: 1863/133

CLOSE SOVIET TIES WITH ROBOTRON

Leningrad LENINGRADSKAYA PRAVDA in Russian 7 Jul 83 p 4

[Excerpt] Yesterday in Leningrad occurred the triumphant delivery of the 300th computer of the "Robotron" combine of the German Democratic Republic, destined for the Soviet Union, for the collective of the Baltic Steamship Line.

One of the largest establishments of the GDR, the combine is well known in the USSR. Its computers, information processing systems and programming facilities are widely used in industrial firms, in public health institutions, in banks, hotels and other institutions of our country.

Head of the steamship line, V. I. Kharchenko, said that use of the Robotron equipment would permit sailors to increase the effectiveness of their work, and would provide additional opportunity for improvement in the quality of control in the transport process.

"Mutually beneficial contacts with our Soviet partners", underlined D. Walter, assistant to the general director of the "Robotron" combine, "enable us to provide the electronics industry of the GDR with long-term orders, accelerate the implementation of new technology, and, as a result, expand the friendship and collaboration between our countries."

On the event of the production of the 300th computer, a reception was held in the General Consulate of the GDR.

We publish the material of the correspondent for the Dresden newspaper SAKSICHE ZEITUNG, Renate Just, about the friendly relations of the "Robotron" combine with the Leningrad plants.

These contacts already have a several year history. Today more than 40 plants in the Dresden area are united by the bonds of partnership with Leningrad establishments related to their work.

To the number of Dresden factories supporting close relations with their Soviet partners belongs the people's factory - the "Robotron" combine. Constructed in 1969, it is occupied in the exploitation, production and

exportation of computer and office technology, and electronic measuring devices. Seventy percent of all the combine's production is exported to 60 countries of the world. The main commercial partners - countries - are memberssof the CEMA, in the midst of which the Soviet Union occupies the foremost place.

Yearly, the export of goods to brother countries steadily increases. This year it rose approximately 14 percent in relation to its previous level. It is the ambition of the thousand workers and employees of "Robotron" to realize delivery of goods by the established deadlines, and to provide for their high quality.

Yet "Robotron" not only supplies a line of products to the USSR, but in return receives from the Soviet Union a variety of devices including tape drive mechanisms, sound recording devices, technical means of program control and minicomputers. Specialists of the combine, jointly with their Soviet colleagues, lead research in the field of development of industrial systems.

In numbers the Soviet Union has been sent its 300th computer, manufactured at its direction. The computer provided is the YeS-1055M. In the computing center of the Baltic Steamship Line it is used in the solution of problems connected with the operational use of vessels, but also in the processing of bookkeeping and statistical data, of pertinent transactions and the technical condition of the fleet.

The great reknown of the apparatus produced by the "Robotron" combine is attested to by the fact that the Baltic Steamship Line has already brought into use a fourth computer with that firm's trademark. To be exact, in 1977 it received from "Robotron" the fiftieth computer dispatched to the Soviet Union. In all, 12 computers now being used in the Leningrad area were supplied by the Dresden firm.

CSO: 1863/180-P

MINICOMPUTERS IN AUTOMATED CONTROL SYSTEMS: ELEKTRONIKA-100-25

Moscow TEKHNIKA I VOORUZHENIYE in Russian No 5, May 83 pp 4-5

[Article by M. Khanov, chief control designer, USSR Ministry of the Electronics Industry, and A. Yeremin, production association department chief]

[Excerpts] Minicomputers of the "Elektronika-100" family have been developed and are already being used for automated control systems in different branches of the national economy. The central place among them is occupied by the high-speed "Elektronika-100-25" minicomputer, which has capabilities exceeding those of the previously produced "Elektronika-100/I" computer. Although they take up the same amount of space, the productivity and computational power of the "Elektronika-100-25" is greater by a factor of 4-5.

Principles previously used in large computers were used in the creation of the "Elektronika-100-25" minicomputer: an increased number of general-purpose registers in the central processor and a multilevel vector interrupt system. It is precisely these principles, which were developed and realized in the "Elektronika-100-25" minicomputer and later in the "Elektronika-79" minicomputer, that determine their high productivity and insure the efficient operation of control complexes in all the basic functioning modes. For instance, the equipment for the hardware processing of input-output inquiries, the superoperational memory and the hardware-software monitoring facilities operate in the real time mode.

The "Elektronika-79" is program-compatible with the more recent computers in the "Elektronika-100" family, as well as "SM-3" and "SM-4" computers.

11746
CSO: 1863/164

UDC 681.32

UNIT FOR INPUT OF DIGITAL INFORMATION INTO 'ELEKTRONIKA TZ-16M' SVM

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA (NAUCHNO-PROIZVODSTVENNYY SBORNIK) in Russian No 4, Oct-Dec 82 (manuscript received after revision 14 Apr 82) pp 47-49

[Article by I.Ye Izvolenskiy and K.V. Konovalov, engineers]

[Text] Industry has mastered the production of the "Elektronika TZ-16M" specialized control minicomputer. The extensive capabilities of this SVM [specialized computer], along with its small size and the simplicity of programming it, make it possible to use it successfully for the automation of various production processes.

Designing information, computation and control systems on the basis of these SVM's is coupled with the specifics of information input into the interface board unit [BIP] that operates in combination with this SVM.

The BIP provides for the input of digital information over one channel and the switching of three analogous ones (instruments of the TsUIP [expansion unknown] type can be used as the information source). The form of the incoming information's representation is a parallel, five-bit, binary-decimal code with a floating decimal [1].

Automated control systems contain different numbers of sensors, with information coming from them most frequently in the form of parallel-sequential or sequential codes. In connection with this there arises the need to develop units for coupling the sensors and switching the information coming from them, with conversion of the information into a form that is suitable for input into the SVM's BIP.

Below we discuss the operating principle of an input information switching unit (BK) that has been developed for the information and computation complex (IVK) of the automated system for the rapid quality analysis (ASEAK) of raw materials that has been introduced at the Chemerskiy alcohol plant.

The ASEAK's IVK is composed of: an "Elektronika TZ-16M" SVM, a BIP, a "Konsul-254 (260)" typewriter, a PL-80/8 perforator, an FS-1501 photoreader and the BK.

Information from the ASEAK's sensors, in the form of a parallel-sequential code, arrives asynchronously, with considerable time lags, for several batches of raw material that has been analyzed [2]. Therefore, the BK must perform the following functions: switch the incoming information; form a channel code number (KNK) for each communication channel; form a batch identification code (KIP) for the information

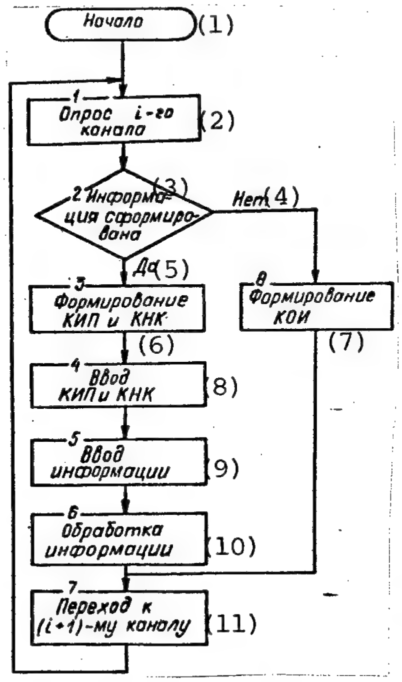


Figure 1. Operating algorithm of IVK's switching unit.

Key:
1. Beginning
2. Interrogate i-th channel
3. Information formulated
4. No
5. Yes
6. KIP and KNK formation
7. KOI formation
8. KIP and KNK input
9. Information input
10. Information processing
11. Switch to (i + 1)-th channel

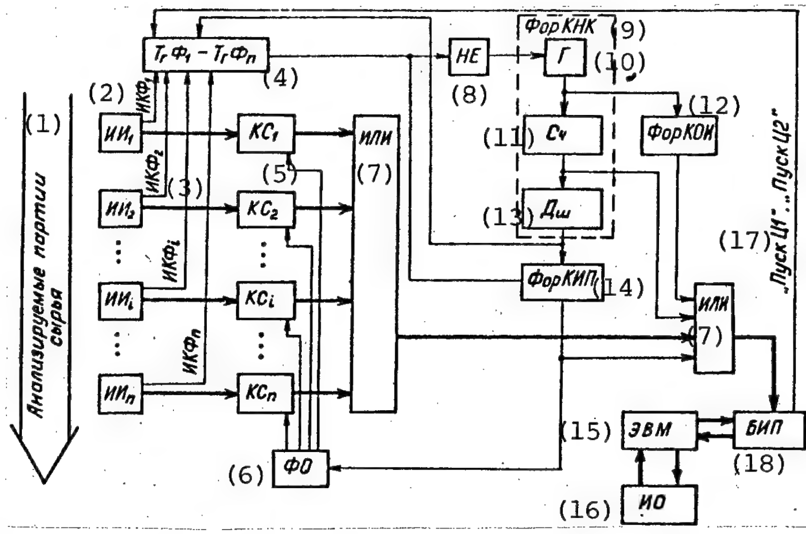


Figure 2. Functional structure of switching unit.

Key:
1. Batches of analyzed raw material
2. II_x (information source)
3. IKF_x (end of forming pulse)
4. TgF_1-T_gF_n (registration trigger)
5. KS_x (communication link)
6. FO (query former)
7. "Or"
8. "Not"
9. ForKNK (KNK former)
10. G (driving oscillator
11. Sch (ring counter)
12. ForKOI (KOI former)
13. Dsh (channel number decoder
14. ForKIP (KIP former)
15. Computer
16. IO (software)
17. "Start cycle x"
18. BIP (interface board unit

arriving over a single channel; form an information absence code (KOI) for channels in which information has not been formulated at the moment of the inquiry.

The operating algorithm of the IVK's BK is shown in Figure 1.

The BK interrogates the communication links continually (operator 1), checking on the appearance of formulated information in the i-th channel (operator 2). In the absence of information in the i-th channel, the BK forms the KOI (operator 8), which realizes the changeover to interrogation of the (i + 1)-th channel (operator 7).

The appearance of information in an interrogated channel causes the formation of KIP and KNK (operator 3) and their entry in the SVM (operator 4). The KIP and KNK determine the address of the register in the SVM's main memory into which the information that has arrived is then entered (operator 5). After the data are processed and printed out (operator 6), the SVM transfers control to the BK, which allows the changeover to interrogation of the (i + 1)-th channel (operator 7).

A channel's serial number is used as its KNK.

The absence code can be any number known not to coincide with a KIP (the number of KIP's in the ASEAK does not exceed the maximum possible number of simultaneously analyzable batches of raw material).

In the case under discussion, KOI = max KIP + 1; that is, it exceeds the possible KIP's by one.

The functional structure of the BK, which realizes the indicated algorithm in combination with the BIP and the SVM, is shown in Figure 2.

The batches of raw materials being analyzed arrive at 2-min intervals and pass through a sample processing line, where they undergo physicochemical analysis.

Upon completion of the transformation of the analog information into digital information, information sources $II_1, II_2, \ldots, II_n$ send formulation completion pulses $IKF_1, IKF_2, \ldots, IKF_n$ into information presence registration triggers TgF_1-TgF_n, thereby changing them into the unit state.

Driving oscillator G of the KNK shaper (ForKNK) sends pulses into ring counter Sch, which is connected to channel number decoder (Dsh). The Dsh pulses interrogate TgF_1-TgF_n in sequence. A zero state of TgF_i allows the passage of the next pulse from G into Sch and, thereby, the interrogation of TgF_i [sic], by simultaneously forming KOI (ForKOI). In connection with this, the BIP forms the signal "Start cycle 1," which allows further interrogation of TgF_1-TgF_n.

If TgF_1 is in the unit state, the generator is slowed down through inhibitor "Not," which thereby prohibits the passage of pulses into the Sch, stopping ForKOI and allowing the formation of KIP (ForKIP). Here the KNK will be the state of the Sch. Both codes are entered in the computer through the "Or" circuit and form the address of the main memory register in which the information will be entered, after which query former FO sends interrogative pulses into communication links $KS_1, \ldots, KS_n$. The information being read passes through the "Or" circuit into the SVM, where it is processed, upon completion of which the SVM's BIP emits a "Start cycle 2" pulse for the reception of information, which triggers the appropriate TgF, after which the cycle starts anew.

The solutions used during the development of the switching unit that realizes these functions were caused by the technical innovations incorporated in the development of the ASEAK's IVK.

The BK's block construction principle makes it possible to put together assemblies depending on the number of communication links.

Series 155 integrated circuits were selected as the element base. All the other BK parts are standard.

The switching unit that was developed has been used in an information and computation complex intended for the collection of information and the calculation of the qualitative characteristics of raw materials arriving for storage. This BK can be used in systems for the automation of experiments that utilize the "Elektronika-70" or "Elektronika TZ-16" SVM's in combination with BIP or BIP 1 interface board units.

Basic Characteristics of Switching Unit

Number of switchable channels	8
Input signal parameters	5 binary-decimal digits
Input and output signal levels, V:	
"0"	no more than +0.4
"1"	no more than +2.5
Supply voltage	220 V ± 15 percent
Power consumption	15 W

The experimental prototype of the BK was manufactured at the experimental plant of the "Pishchepromavtomatika" VNPO [possibly All-Union Scientific Production Association] in Odessa.

BIBLIOGRAPHY

1. Chaplygin, A.B., Vedernikov, M.V., and Burkov, A.T., "Coupling the Small 'Elektronika-70' and 'Elektronika TZ-16' Computers to Peripheral Gear," PRIBORY I TEKHNIKA EKSPERIMENTA, No 4, 1978, pp 85-87.

2. Konovalov, K.V., Izvolenskiy, I.Ye., Flon, V.S., and Lutsyk, V.I., "Computer and Information Complex of an Automated System for the Rapid Analysis of Raw Material Quality," MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA, No 4, 1979, pp 28-31.

11746
CSO: 1863/85

UDC 681.322

STRUCTURE OF DATA PROCESSING SUBSYSTEM OF MULTIPROCESSOR COMPUTER SYSTEMS WITH REARRANGEMENT STRUCTURE

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83 (manuscript received 4 Mar 82) pp 3-6

[Article by A.A. Zabolotnyy, V.V. Ignatushchenko, V.M. Kostelyanskiy, G.M. Lekhnova and D.A. Nedzel'skiy, "Selection of the Structure of a Data Processing Subsystem of Multiprocessor Computer Systems with a Rearrangeable Structure", in the section "Computer Resources and Support Equipment of Systems]

[Text] A promising new feature in the development of multiprocessor computer systems (MVS) is creating them with a rearrangeable structure, as described in [1]. The distinguishing features of such systems are their rearrangeability and the presence of both scalar and vector instructions in the architecture. By rearrangeability is meant the system's ability to allocate to the process served the computer resource required at a given point in time. The system realizes the processing principles "many instruction flows--many data flows". The number of instruction flows corresponds to the number of control units (UU), and defines the number of simultaneously solved tasks and subtasks of a task.

The features of an MVS such as the one described in [1] impose specific requirements on the data processing subsystem's structure. In conventional computer systems (both single- and multiple-processor ones), each processor, in executing one instruction flow, generally uses only its own computing resource (since all the instructions are scalar). The main requirement on such a data processing subsystem is that it ensure the minimum possible complete data processing time.

To ensure a low time for complete processing of scalar operands in high-performance computer systems, due to the low effective depth of the instruction pipeline (1.6-4, according to the data in [2]), the processing subsystem consists of several special-purpose devices (for processing floating-point numbers, fixed-point numbers, decimal data, etc.) of a combination (where the action is executed in one cycle) or pipeline type, but with a low pipeline depth. Such devices enable fast processing of operands, but require large equipment costs for realization, since all the actions are performed in parallel and considering the worst combinations. Since the DP subsystem is generally used by one instruction flow, the coefficient of its use is low, especially in view of the effect of various factors causing instruction pipeline interruptions.

In processing vector instructions, the main requirement on the DP subsystem is to ensure a high throughput, even at a considerable total processing time of one pair of operand elements, since the vector operands have dimensionalities l considerably exceeding the pipeline depth when processing scalar instructions ($l \gg 1.6-4$). Consequently, the vector subsystem of processing can be built either of n similar processors (whose complete processing time requirements may be not very high at considerable n), or of j special-purpose pipeline type devices, but with a rather large pipeline depth. The design of the processing subsystem of an MVS with a rearrangeable structure of type [1] must therefore efficiently ensure the following characteristics: a use coefficient of the processing subsystem close to one; maximum throughput when processing vector operands, regardless of the complete processing time of one pair of vector operand elements; and minimum complete processing time of scalar operands.

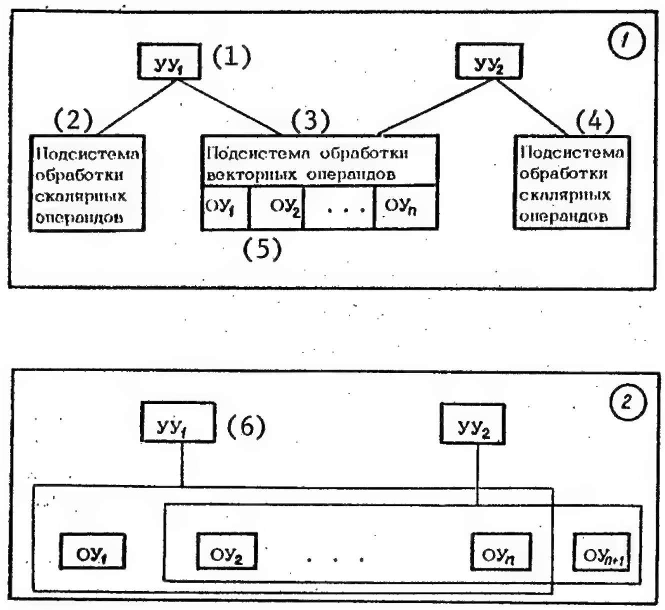
The main way to raise the use coefficient of processing devices is to organize the total computer resource (OVR) for several instruction flows [3]. They should be organized for not more than four instruction flows.

To ensure the minimum complete processing time (and consequently, a high throughput), the OVR must be organized of j special-purpose devices, executing actions in one or several cycles. However, this is not an efficient way in terms of machine costs. In a type [1] MVS there is a large set of data formats (from a byte to a 65-digit floating-point); therefore, the set of special-purpose devices is also considerable. For executing floating-point operations, the devices must be optimized to process the maximum data format (65 digits). Much of the equipment is not used when processing smaller data formats.

It is therefore preferable to organize the OVR of n universal similar devices (n must not be large, to ensure a sufficient universality of the system [2]), which are adjusted to process data of the most probable format (for example, for processing 32-bit operands with a floating point). However, such devices must enable adequate processing of other data formats as well, without substantial additional machine costs. In this case, all n devices participate in processing vector instructions, and $i < n$ devices in processing scalar instructions (n-i devices are idle, resulting in a drop in the processing subsystem's use coefficient).

Dynamic redistribution of the processing devices at each computation stage enables elimination of equipment down times, but requires additional equipment costs for UU and processing device switching. Parallel execution of operations on vector and scalar data are provided in dynamic redistribution of processing devices; however, the maximum performance of the processing subsystem is limited to the UU throughput (one pair of operands per cycle) and the communication channels between the UU and the processing devices.

A simpler and more efficient method of enhancing the processing subsystem's use factor is to divide it into two parts (Fig. 1): one for processing vector operands (OVR for two UU's); the other, for processing scalar operands (a separate resource for each UU). In this case, the machine costs rise somewhat, but parallel execution is possible of both scalar and vector operands, which makes it possible to avoid reducing the use factor of both the UU and the processing subsystem as a whole.



Figs. 1 and 2. Division of the Processing Subsystem into Two Parts

Key.
1. Control unit
2. Scalar operand processing subsystem
3. Vector operand processing subsystem
4. Scalar operand processing subsystem
5. Operating unit
6. Control unit

Another variant of dividing the processing subsystem into two parts is given in Fig 2, where the OVR's for two UU's are OU_2-OU_n (OU: operating unit)

To process vector operands, UU_1 uses OU_1-OU_n; UU_2, OU_2-OU_{n+1}. To process scalar operands, OU_1 and OU_{n+1}, respectively. In the system given in Fig. 2, the equipment costs are somewhat less than in Fig. 1's system, but simultaneous execution of scalar and vector instructions is possible.

During sequential execution of scalar and vector instructions on one machine, time loss for correcting addresses and operands, checking conditions and organizing cycles is inevitable even with considerable dimensions of the vector operands [4], resulting in a drop in the entire system's performance.

Let us compare the two versions (refer to Figs. 1 and 2) of the processing subsystem from the standpoint of their performance given comparable machine costs. We shall assume that their throughputs during processing of vector and scalar operands are equal.

Let us evaluate a rise in subsystem performance in Fig. 1 compared with the subsystem in Fig. 2 by combination (parallel processing) of the scalar and vector instructions.

Assume that both versions of subsystem processing process the same instruction

flow. Any program written according to the architecture of the type [1] MVS is divided into fragments, containing only vector and only scalar instructions (referred to as vector and scalar fragments, respectively). After the vector fragment must be a scalar one, and vice versa. We shall represent our system, consisting of one UU and a processing subsystem, as a two-phase mass service system without inquiry buffering.

The first phase (UU) generates program fragments; the second phase (processing subsystem) executes. We shall also assume that the distribution of generation and execution time of vector fragments and processing time of scalar fragments has an exponential law with intensities μ, λ, γ, respectively (by processing of scalar fragments is meant both their generation and execution). The presence of the second UU in the system will be taken into consideration using the probability P of conflict for the OVR.

We shall define the probability of OVR assignment for the system given in Fig. 2 as:

$$P = 1/\lambda : (1/\mu + 1/\lambda + 1/\gamma). \quad (1)$$

Selection of the exponential law of distribution of times of generation and execution of segments gives us low estimates of system performance, and a low boundary of the gain in performance by combination. Given any other laws, the estimates will be better [5].

The behavior of the system in Fig. 2 can be described using a Markov chain, whose graph of states is given in Fig. 3.

The states of the system will be described as follows: a_0, the UU generates the vector fragment with intensity μ; a_1, the OVR executes the vector fragment with intensity λ; a_2, the UU is blocked, such that the OVR is engaged in executing the vector fragment of the other UU; a_3, the UU and part of the OVR (OU_1 or OU_{n+1}) execute the scalar fragment with intensity λ.

After completion of execution of the vector and scalar fragments, the system returns to the initial state a_0.

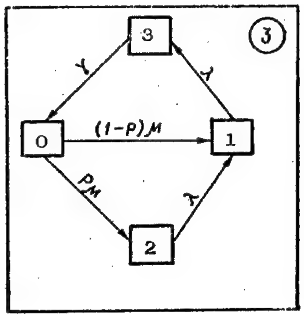
The time for return to the initial state

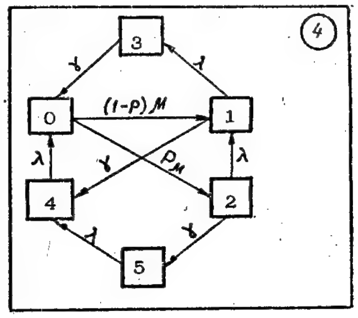
$$T_1 = 1/\mu + 1/\lambda + 1/\gamma + P/\lambda, \quad (2)$$

where P is defined from (1).

The behavior of the system of parallel processing of vector and scalar fragments (see Fig. 1) can be described using a Markov chain, whose graph of states is given in Fig. 4.

The states of the system will be described as follows: a_0, the UU generates the vector fragment; a_1, the OVR executes the vector fragment, UU_1 and OU_1 process the scalar fragment; a_2, UU_1 and OU_1 process the scalar fragment, OVR executes the vector fragment of the other UU; a_3, UU_1 and OU_1 process the scalar fragment; a_4, OVR executes the vector fragment, the scalar fragment is already executed (UU_1 and OU_1 are idle); a_5, the scalar fragment is executed, UU_1 is blocked, OVR executes the vector fragment of the other UU.





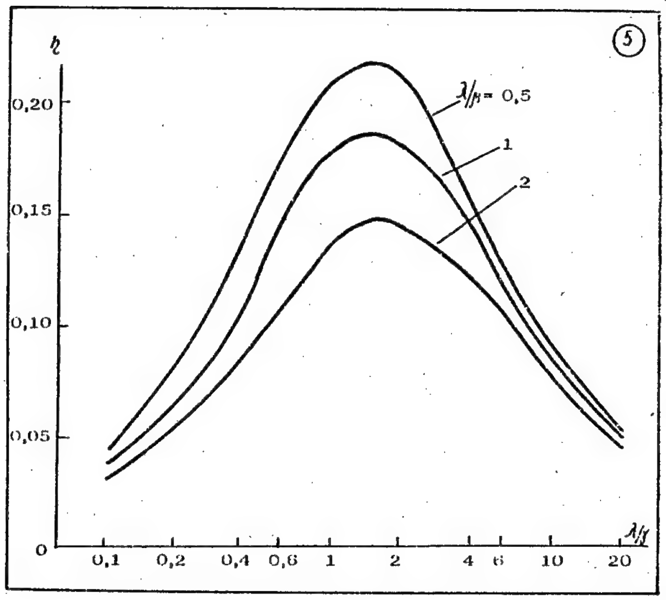


Fig. 3. Graph of States of Fig. 2's System

Fig. 4. Graph of States of a System of Parallel Processing of Vector and Scalar Fragments

Fig. 5. Dependences of $\eta=(T_1-T_2)/T_1$ on various system parameters

It can be shown that the time for return to the initial state

$$T_2 = 1/\mu + 1/\lambda + 1/\gamma + P/\lambda - P\lambda/(\lambda+\gamma)^2 - 1/\lambda + \gamma. \quad (3)$$

Fig. 5 gives the dependences of $\eta = (T_1 - T_2)/T_1$ on various system parameters.

It follows from Fig. 5 that the maximum gain from combining processing of scalar and vector fragments is reached at $\lambda/\mu \simeq 1.5$ for various values of λ/μ. Given any deviation from the extremum point, the gain is considerably less. It is assumed in studying Fig. 4's model that combination of processing of the scalar and vector fragments is

always possible. However, when executing actual programs, there can exist, for instance, an information dependence between the scalar and vector instructions, reducing the possibility of their combination. The operand addresses are checked to ensure combination of the processing of several instructions.

In conventional computer systems, all the instructions are scalar; therefore, addresses of only the scalar operators need be checked; for example, i addresses at a depth of combination i. According to the data in [2], the depth of combination lies in the range of 1.6-4; therefore, not more than 4-8 addresses must be checked.

The situation in MVS's with vector instructions is considerably more complicated. For example, to check the possibility of combining the scalar instruction with the vector instruction (given the condition that the step of the vector operands is a single one), it is sufficient to compare the address of the scalar operand with the initial and final addresses of the vector operand. This already requires more equipment than in checking the scalar operands (since the check uses adders, not the associative memory), and additional time spent on forming the final vector address. When the vector step is not a single one, the comparison of the scalar operand with the initial and final addresses of the vector does not ensure a final determination of the information dependence, and organizing the combination requires an even greater machine cost.

At the same time, an analysis of the programs shows that a considerable part of the scalar instructions executes preparatory actions for the vector instructions. These scalar instructions include address correction, counter modification and check, formation of data for the next cycles, etc. Vector instructions of processing and organizing scalar instructions belong to different instruction groups (according to the type [1] MVS architecture). Consequently, there is not an information dependence between them, and they can be executed in parallel, without a considerable machine cost for checking the possibility of combination.

The necessity of organizing combination of vector and scalar processing instructions is defined by the performance requirements of the processing subsystem, allowable machine costs, and type of programs run. If the processing scalar instructions comprise a minor part of all the scalar instructions, their combination with the vector instructions yields a minor gain (see Fig. 5 at $\gamma >> \lambda$), at considerable machine cost.

Moreover, combination of the vector processing instructions and organizing scalar instructions should be organized regardless of the relation between γ and λ, since a check of the possibility of such a combination does not require additional machine costs, while the combination enhances system performance.

In conclusion, we can draw the following conclusions.

In MVS's with a rearrangeable structure of type [1], the data processing subsystem should be divided into two independent and parallel-operating parts: one, for processing scalar operands; the other, for processing vector operands.

The maximum effect from combining the vector and scalar instructions is achieved at close values in intensities of execution of these instructions ($\gamma \sim \lambda$) in a wide

range of values of intensities of generation (μ) and execution (λ) of vector instructions.

Combining vector instructions of processing and organizing scalar instructions is advisable regardless of the relation between intensities of execution of these instructions.

The advisability of combining vector and scalar instructions of processing is defined by the type of programs executed, requirements on system performance and allowable equipment costs.

To ensure values of the use coefficient close to one for the resources processing the vector operands, it is advisable to make them common to several control units. The resources executing the scalar instructions must be separate for each UU.

BIBLIOGRAPHY

1. Prangishvili, I.V., Rezanov, V.V., "Mnogoprotsessornyye upravlyayushchiye vychislitel'nyye kompleksy s perestraivayemoy strukturoy" [Multiprocessor Control Computer Complexes with a Rearrangeable Structure], Moscow, 1977, 25 pages (ITM and VT AN SSSR, preprint No 10).

2. Kartsev, M.A., Brik, V.A., "Vychislitel'nyye sistemy i sinkhronnaya arifmetika", [Computer Systems and Synchronous Arithmetic], Moscow, Radio i svyaz', 1981, 360 pages.

3. Zabolotnyy, A.A., "An Analysis of the Organization of Common Computer Resources in Uniform MVS's", AVTOMATIKA I TELEMEKHANIKA, 1981, No 2, pp 164-172.

4. Zabolotnyy, A.A., Nedzel'skiy, D.A., "Analysis of Processing Vector Operands by Fragments", in "Problemy sozdaniya i ispol'zovaniya vysokoproizvoditel'nykh informatsionno-vychislitel'nykh mashin: Tez. dokl. vsesoyuz. nauchno-tekhn. sov." [Problems of Creating and Using High-Performance Information Computers: Papers of the All-Union Scientific and Technical Conference], Kishinev, 1979, pp 21-23.

5. "Osnovy teorii vychislitel'nykh sistem" [Foundations of Computer System Theory], edited by S.A. Mayorov, Moscow, Vysshaya shkola, 1978, 408 pages.

9875
CSO: 1863/140

UDC 681.325.33:519.712.5

UNIVERSALITY OF DISCRETE SIGNAL CONVERSION, GENERATION AND RECEPTION DEVICES

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83
(manuscript received 22 Feb 82) pp 7-10

[Article by S.M. Krylov, "The Universality of Discrete Signal Conversion, Generation and Reception Devices"]

[Text] A considerable amount of interest is currently being shown towards territorially distributed systems of monitoring and control based on microprocessors and micro- and minicomputers. An important function in such systems is performed by the communication networks, providing transmission of information between data collection subsystems, peripheral processors, CPU's and executing devices. Such networks are usually based on existing communications, including telephone, telegraph and other communication channels, using various data transmission protocols. Moreover, in some cases system efficiency requires the use of specially-developed protocols; for example, a two-line version of the "Common Bus" interface, start-stop protocols with a nonstandard length of the word transmitted, etc. Similar problems arise in assigning digital computers to various memories, using the principle of sequential writing of the information on a magnetic medium.

It would be difficult to cover the entire range of existing and theoretically possible protocols for transformation of information from a parallel form of representation to a sequential one and back using existing equipment such as communication adapters [1,2]. It would be more efficient and (from the engineering standpoint) more promising to create universal programmable devices [3] for such purposes, realizing any data conversion laws. This would obviously allow maximum standardization of the corresponding hardware.

This article proposes a model of a universal programmable communication device, and considers certain aspects of its practical embodiment.

The model's structure is given in Fig. 1. The model (denoted M) contains a shift register (RS) with a pointer: a unit in one of the digits; zeros are written in the remaining RS digits. The information register (RI) consists of a linear array of flip-flops. Reading and writing into them is done by the instructions "Read x by the Pointer", "Write x by the Pointer". The binary signal read from the RI is written into flip-flop T_S, whose output is connected through the controlled matching amplifier A2 to the input of the communication line, whose output, in turn, is connected through the threshold detector A1 with the data write input in the RI. The signal at the A1 output is shaped according to the law:

$$P(y) = \begin{cases} 1, & \text{if } y \geq 0, \\ 0, & \text{if } y < 0. \end{cases}$$

If amplifier A2's work is authorized, the signal y at its output is defined by the conditions:

$$y = \begin{cases} +A, & \text{if in } T_s \text{ is written } 1, \\ -A, & \text{if in } T_s \text{ is written } 0, \end{cases}$$

where +A and -A are the positive and negative levels of the discrete signal generated into the communication line.

The model M also contains a programmable time counter: the counter Tm, controlled using the instructions: "Load Into Timer Number K", "Start". Upon the "Start" instruction, the timer begins operating in the mode of subtraction (at a fixed frequency) of units from the number loaded in it. The control unit can define the timer's status by the instruction: "If the timer contents are greater than zero, go to P; otherwise, execute the next instruction" (P is the initial address of the subroutine in the memory device of the instructions to which the jump occurs if the timer contents do not equal zero). When the "zero" state is attained, the timer stops automatically. Moreover, M executes the instructions: "Write 1 by Pointer", "Write 0 by Pointer", "Shift Pointer by 1 Digit to the Right (Upwards)", "Shift Pointer by 1 Digit to the Left (Down)", "Unconditional Jump to P", "Stop", "If 1 is written in the RI flip-flop the pointer indicates, go to P; otherwise, execute the next instruction". It also indexes the forbidden status, where the pointer is in the extreme low position and the "Shift Pointer to the Left" instruction must be executed.

It is easily shown that the structure shown in Fig. 1 is universal in the Turing sense; i.e., any algorithm realized in Turing machines can be realized in it. In fact, for a two-symbol Turing machine with an alphabet $\Sigma = \{0, 1\}$, equivalent operations can be indicated in M if the M information register is interpreted as a tape of the Turing machine; flip-flops $T_0, T_1, \ldots, T_i$ as cells of the tape; the states of the flip-flops as symbols of 0, 1; the pointer a_i, as the read/write head; and reading and discrimination of symbols are done using a sequence of two instructions: "If 1 is written in the RI flip-flop the pointer indicates, go to P1; otherwise, execute the next instruction"; and "Unconditional jump to P2", where P1 and P2 specify the addresses of the subroutines corresponding to the states of the Turing machine to which they jump during similar reading of a similar symbol in the tape cells.

It is easy to go from a two-symbol Turing machine using encoding/decoding algorithms to a machine with any finite number of symbols [4].

Let N be a set of natural numbers. It follows from the universality of M that for any n, $m \in N$ in M can compute the system of functions:

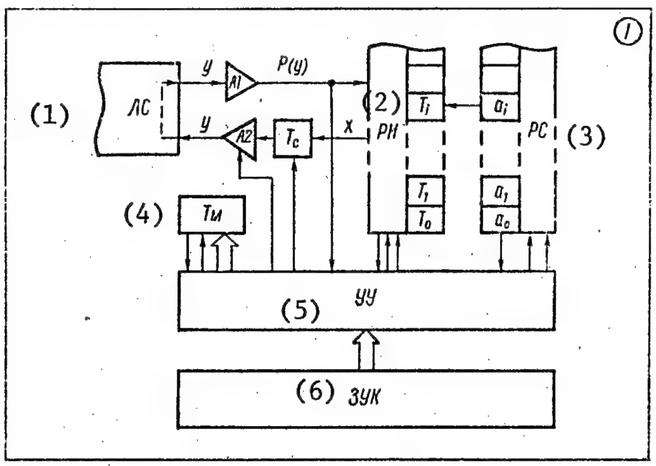


Fig. 1. Structure of the Model M of a Universal Communication Processor

Key:

1. Communication line
2. Information register
3. Shift register
4. Timer
5. Control unit
6. Memory device of instructions

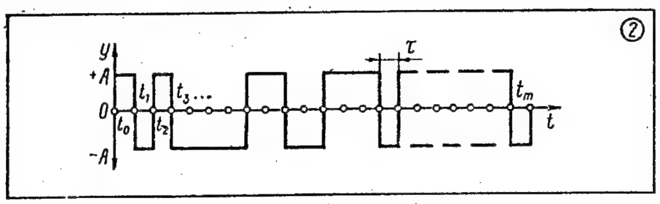


Fig. 2. Diagram of a Hypothetical Data Transmission Protocol

$$\left.\begin{array}{l} y_0 = f_0(x_0, x_1, \ldots, x_n), \\ y_1 = f_1(x_0, x_1, \ldots, x_n), \\ \vdots \\ y_m = f_m(x_0, x_1, \ldots, x_n), \end{array}\right\} \qquad (1)$$

where y_i, $x_j \in N$; f_k are general recursive functions [5].

Let us consider the diagram in Fig. 2, illustrating a certain hypothetical data transmission protocol. The protocol is a sequence of (m+1) elementary signal elements of length τ, whose values (+A or -A) at corresponding intervals of time t_0, $t_1, \ldots, t_m$ are computed by the formulas:

(if)

$$
\left.\begin{aligned}
y(t_0) &= \begin{cases} +A, & \text{если } y_0 = f_0(x_0, \dots, x_n) = 1, \\ -A, & \text{если } y_0 = f_0(x_0, \dots, x_n) = 0; \end{cases} \\
y(t_1) &= \begin{cases} +A, & \text{если } y_1 = f_1(x_0, \dots, x_n) = 1, \\ -A, & \text{если } y_1 = f_1(x_0, \dots, x_n) = 0; \end{cases} \\
y(t_m) &= \begin{cases} +A, & \text{если } y_m = f_m(x_0, \dots, x_n) = 1, \\ -A, & \text{если } y_m = f_m(x_0, \dots, x_n) = 0 \end{cases}
\end{aligned}\right\} \quad (2)
$$

Obviously, if the values $y_0,\dots,y_m$ are known and are located, for example, in flip-flops $T_{n+1}-T_{n+m+2}$, while the interval τ can be counted off by the timer M, then the signals $u(t_0),\dots,y(t_m)$ can be synthesized in M by sequential rewriting into the flip-flop T_s of the values of flip-flops $T_{n+1}-T_{n+m+2}$ at identical intervals τ, counted by the timer Tm (work of the amplifier A2 must be allowed by the appropriate signal).

On the other hand, functions of the form $y_i=f_i(x_0,\dots,x_n)$ form system (1); i.e., if the functions f_0-f_m are computable, then any transmission protocol obtained from (1) in (2) can be realized in M.

It is easily seen that with the appropriate selection of τ it is possible to ascribe to form (2) all final protocols and laws of information conversion from a parallel form of representation (binary code x_o-x_n in flip-flops T_0-T_n) into a sequential one. For example, system (2) can describe code-pulse modulation, frequency modulation (for a square carrier), relative phase keying (for a square carrier), signals of the set of the user's number, conversion of information using Hamming codes, and so forth.

Fig. 3 shows the commonly known start-stop protocol with a start element (x_0), five information elements (x_1-x_5), and control (x_6) and stop (x_7) elements, for which:

$$
\left.\begin{aligned}
y_0 = y_1 = x_0 &= 0, \\
y_2 = y_3 &= x_1, \\
y_4 = y_5 &= x_2, \\
y_6 = y_7 &= x_3, \\
y_8 = y_9 &= x_4, \\
y_{10} = y_{11} &= x_5, \\
y_{12} = y_{13} = x_6 &= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5, \\
y_{14} = y_{15} = y_{16} = x_7 &= 1,
\end{aligned}\right\} \quad (3)
$$

where $x_1,\dots,x_5$ are binary single-digit codes of the corresponding elements of a five-digit parallel code $x_1,x_2,\dots,x_5$; $\oplus$ is the symbol of the modulo 2 addition.

System 3 is computable, corresponds to form (1), and is consequently realizable in M. Moreover, it is not difficult to construct a function F_U computable in M that encompasses all possible final protocols and laws of conversion of binary data from a parallel to a sequential form. The known property of the Gödel function $\Gamma(z,i)$ [5] can be used for this, according to which for any finite sequence $y_0, y_1,\dots,y_m$ there is a parameter z', when

$$
\left.\begin{aligned}
y_0 &= \Gamma(z', 0),\\
y_1 &= \Gamma(z', 1),\\
&\vdots\\
y_m &= \Gamma(z', m).
\end{aligned}\right\} \tag{4}
$$

Assuming that y_i can take the values of only 0 or 1, for a given protocol A_i and given binary code $x_0,\ldots,x_n$ it is easy to compute the values of $y_0,\ldots,y_m$ of system (4), solving which we find the value sought of the parameter z'. Obviously, such parameters exist for other code combinations of elements x_0, $x_1,\ldots,x_n$ as well. Arranging these parameters in the lexicographic order z'_0, $z'_1,\ldots,z'_s$, where $x=2^n-1$, and parameter z'_i corresponds to the natural number i, having in the binary recording the code $x_0,x_1,\ldots,x_n$ $(x_j \in \{0,1\})$, we obtain the final sequence for which there is also a parameter z", such that

$$
z'_i = \Gamma(z'', i),\ i \in \{0,1,\ldots,(2^n-1)\}\ . \tag{5}
$$

With (4) or (5) written together, we have:

$$
\left.\begin{aligned}
y_0 &= \Gamma(\Gamma(z'', i), 0),\\
y_1 &= \Gamma(\Gamma(z'', i), 1),\\
&\vdots\\
y_m &= \Gamma(\Gamma(z'', i), m)
\end{aligned}\right\} \tag{6}
$$

or

$$
y_j = \Gamma(\Gamma(z'', i), j),\ j = 0, 1), \ldots, m. \tag{6a}
$$

The right-hand part of (6a) is the universal function F_U sought, giving for the specified type of protocol, defined by the parameter z", and the specified binary code, defined by the parameter i, the values $y_0,y_1,\ldots,y_m$. Due to the properties of the Gödel function, expression (6) includes not only all possible final protocols that can be written in form (2), but also other types of data conversion; in particular, conversions in which the elements of y_j are not 0 and 1.

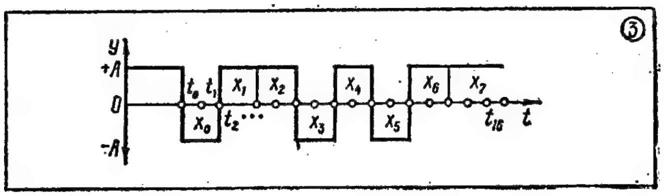
From the preceding discussion, the fact follows rather obviously that by solving system (1) with respect to information elements x_0-x_n; i.e., obtaining the system

$$
\left.\begin{aligned}
x_0 &= f'_0(y_0, y_1, \ldots, y_m),\\
x_1 &= f'_1(y_0, y_1, \ldots, y_m),\\
&\vdots\\
x_n &= f'_n(y_0, y_1, \ldots, y_m).
\end{aligned}\right\} \tag{7}
$$

we also obtain the possibility for inverse conversion in m of sequential messages arriving through the communication line to the parallel code $x_0,x_1,\ldots,x_n$.

In fact, using comparator A1 and timer Tm to question the status of the line in the middle of each interval $t_0,t_1,\ldots,t_m$, it is possible to form in flip-flops T_0-T_m of the information register the sequence $y_0,y_1,\ldots,y_m$. Based on it and

system (7), it is possible to form in flip-flops $T_{m+1}-T_{m+n+2}$ the digits of the parallel code $x_0 \ldots x_n$.



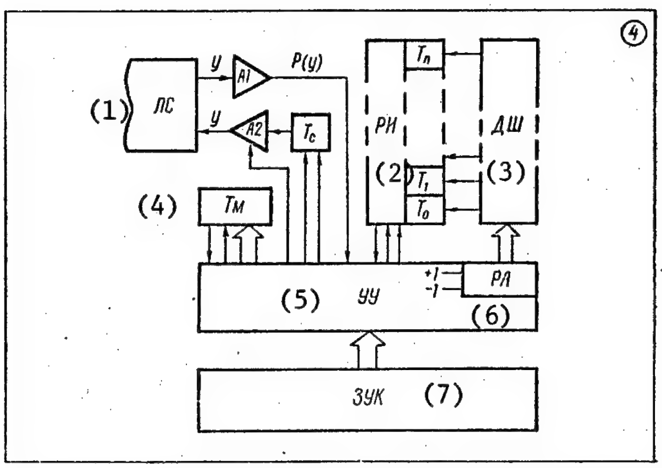


Fig. 3. Start-stop Protocol

Fig. 4. Practical Realization of Model M

Key:
1. Communication line
2. Information register
3. decoder
4. Timer
5. Control unit
6. Address register
7. Memory device of instructions

Questions of synchronization; i.e., determining the initial element of the sequence y_0, generally represent a rather large, independent problem, and are not considered here. We will only note that if there exists an algorithm (i.e., a method) for defining the initial element y_0, then it is realizable in M because of its universality. On the other hand, synchronization in most protocols used in practice (for example, that shown in Fig. 3) does not represent a problem and is done by an elementary method: after each stop element, the start element is defined by switching the signal in the line from level +A to level -A; i.e., the beginning of the message is always the moment of a certain switching of the signal in the line, given the condition that the stop element was previously allocated or the line was known to be in a nonworking status.

In actual systems, a natural limit compared with the model in Fig. 1 will be the length of the information register (Fig. 4). However, it does not fundamentally affect the statements proven above, since for any finite conversion protocol the required information register volume is finite, and can always be achieved. Moreover, practice demonstrates that all the basic data conversion protocols of byte length (and less, naturally) can be realized in a system containing only 20-30 type T_i flip-flops [6].

Another difference between model M and the practical circuit shown in Fig. 4 is the absence of a shift register and a pointer, whose functions are performed by the address register (RA) with the decoder (DSh). To preserve equivalence between Fig. 1's model and Fig. 4's structure, it is sufficient to provide the possibility of changing the address of the current reference to the array of flip-flops of the information register by ± 1. In addition, in place of direct rewriting of information from the RI to T_s, one can use a setting of T_s by UU signals depending on the status, read using the operation of the conditional jump, of the chosen flip-flop T_i of the RI array.

For convenient programming and formation of input, output, auxiliary and service signals, it is advisable to allocate in the RI array several system registers: a register of transmitted information; register of received information; register of control signals; register of internal auxiliary flags, etc. The mechanism of reference to register digits can be modified compared with the M mechanism, but with mandatory preservation of the efficiency of reference; i.e., the possibility of program writing and reading of data from any register digit.

For program reference to various control and synchro signals, it is useful to include in the device's instruction system instructions of conditional jump by external sentinels of the type: "If the i-th external signal equals one, then go to P; otherwise, execute the next instruction".

Based on Fig. 4's structure, several multiple-protocol programmable communication processors have been developed [6], costing about 35-50 SSI's and MSI's. Calculated in terms of single-input gates, this yields equipment costs at the 800-1500 element level; i.e., about the same as required for communication adapters on conventional circuits [2], and considerably less than multiple-protocol communication systems based on microprocessors.

Building devices based on modern TTL-SIS enables speeds of $3 \cdot 10^6$; based on CMOS-SIS, $0.5 \cdot 10^6$ operations per second. In other words, at a data transfer rate up to 10000 baud, in the processing time of a unit of transmitted information the TTL version can execute on the order of $3 \cdot 10^2$ operations; the CMOS version, on the order of 50, which is fully sufficient for calculating relatively complex expressions of the type (2).

We should note in passing a general method of raising the speed of model M when solving the concrete classes of problems written in the form (2). The method consists of finding the most frequently encountered algorithmic constructions and realizing them by the hardware. Such constructions include algorithms of reference to a real-time scale (i.e., operations with a timer Tm), operations of modulo 2 summation of sent-received bits, etc.

Given a large speed reserve, the device can execute other functions in parallel: service additional communication channels; solve interface or service problems; check the equipment.

The variants of the device developed enable formulation of signals for writing data on magnetic media and signals for selecting the user's number for a telephone line. A certain equivalent is also proposed of a "Common bus" type interface for a two-wire communication line, using the special protocol LK81. It enables connection of a theoretically unlimited number of low-speed data transponders to the line.

The range of issues considered provides a justification for defining M as a model of universal, programmable communication processors of a low hierarchical level, primarily designed for relatively low-speed local networks. Construction of a similar universal model for higher-level processors, realizing complex packet communication protocols of the X.25 and HDLC type, requires developing the M structure in the direction of introducing means for parallel data processing, since the data transfer rates in such networks are at least an order higher than those in existing realizations of M.

BIBLIOGRAPHY

1. Trevi, S., "A Communication Adapter that can be Interfaced With Most Microprocessors", ELEKTRONIKA, 1978, No 6, pp 45-53.

2. Kogan, A.L., Kolosovskiy, A.V., Panov, Yu.N., Sinekayev, V.V., "The LIS of the Asynchronous Transponder K1000KhL1", ELEKTRONNAYA PROMYSHLENNOST', 1982, No 2, pp 48-51.

3. Fisher, Dzh.L., "Programmable Components: The Future of SBIS", ELEKTRONIKA, 1980, No 13, pp 47-53.

4. Kuznetsov, O.P., Adel'son-Vel'skiy, G.M., "Diskretnaya matematika dlya inzhenera" [Discrete Mathematics for the Engineer], Moscow, Energiya, 1980, 344 pages.

5. Mal'tsev, A.I., "Algoritmy i rekursivnyye funktsii" [Algorithms and Recursive Functions], Moscow, Nauka, 1965, 392 pages.

6. Krylov, S.M., Shipilov, S.G., "Programmiryemyye protsessory svyazi dlya mini- i mikroEVM" [Programmable Communication Processors for Mini- and Microcomputers], Kuybyshev, KuybTsNTI, 1981, 2 pages (Inform. sheet/KuybTsNTI, No 609).

7. "Mikroprotsessory--Communication Controllers (Report)", ELEKTRONIKA, 1978, No 8, pp 11-12.

9875
CSO: 1863/140

UDC 681.326.65

IMITATION METHOD OF DETERMINING ERROR CORRECTION EFFICIENCY IN DISCRETE COMPUTER OBJECTS BY HARDWARE CHECK MEANS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83 (manuscript received 14 Oct 80, after revision 4 Aug 81) pp 14-17

[Article by B.Ye. Liokumovich, "The Imitation Method of Determining Error Correction Efficiency in Discrete Computer Objects by Hardware Check Means"]

[Text] In the process of designing hardware check facilities (SAK) of discrete objects (DO) of computers, the task arises of determining the efficiency of error correction by different check variants for selecting the check facilities to be added to the object. The effectiveness of error correction ρ will be described by the probabilities of correction and detection of errors arising in the information converted by the DO. Methodologically, the task of defining ρ and the structure of errors (for example, error frequency) has been considered most completely in [1], in our opinion; methods of solving these problems are given in [2-4]. However, these problems have not yet been solved completely, especially that of determining ρ in the process of DO design and given errors caused by equipment malfunctions.

This article considers an imitation method of determining ρ and the structure of errors given failures and malfunctions of DO elements, based on studying DO operation with alternative versions of checking using a model of a discrete object (MDO) with probability simulation of the flow of failures and malfunctions. Along the lines of [5], we shall consider a failure, compared with a malfunction, to be a considerably longer equipment problem.

In the general form, we formulate this problem as follows. In the logical structure of the DO are defined the probable levels and possible methods of information check, and the total number is set of possible versions of check β ($\beta=\overline{1, n_{vk}}$), where n_{vk} is the number of variants. It is necessary to define ρ_β by each variant and check circuit (SK), along with the structure of errors at the interesting levels of check. By level of check is meant the totality of DO digits checked by one SK; by check variant, the totality of SK's of nonrepeating levels, used to check the data.

A correct study of the effectiveness of error correction by SAK assumes definition of the probabilities of detection and correction by the circuits and check versions of errors both in the DO as a whole, and in the checked part of the basic equipment and the control equipment, along with determination of the probability of error in the unchecked part of the basic equipment. The totality of these probabilities

enables analysis of the effect of each SK on the amount of ρ_β, and establishment of the cause for an unsatisfactory value of this criterion.

Given a sufficiently large number of simulatable errors, the probabilities sought of correction (detection, nondetection) of errors in the indicated parts of the DO equipment equal the quotient from dividing the number of correspondingly corrected (detected, undetected) errors by the total number of errors in these equipment parts. The problem of determining the probabilities is therefore reduced to defining the number of corrected, detected and undetected errors in the basic and check DO equipment, from the total number of simulated problems of the object.

The necessity for this degree of detail in solving the problem has defined the following basic requirements that the MDO used must satisfy: reflect the information status of the object elements and intensity of transitions of the elements from correct, operating status to incorrect, and back; include an apparatus for imitating the length of the fault; identify elements by effect on the check levels; and include facilities for simulating check circuits and variants.

The basis of this method is imitation of the operation of the check circuits and variants studied in models of working (model M) and faulty (model M*) DO's, and comparing the states of DO check points given identical signals at the external inputs of the models.

The structural diagram of the problem's solution is given in Fig. 1, where tv,i, tz,i are the time of appearance and completion of fault i in the DO, counted from the beginning of the operating cycle; t_χ is the time of the beginning of cycle χ. Since consideration of the formation principles of the MDO goes outside the scope of this work, we shall here give only the information structure of the model of element d_e in the model of the object:

$$d_e = \{y_e = f_e(X);\ d_{oo};\ d_{ko};\ d_{y,1}(1 = \overline{1,\ n_y})\},\qquad (1)$$

where y_e is the value of the output variable of the element; $f_e(X)$ is the function performed by the element; n_y, the number of check levels; the components y_e, d_{oo}, d_{ko}, $d_{y,1}$ are symbols of the binary alphabet (0,1).

If the component equals one, then y_e is the flag of the logic unit at the output of the element; d_{oo}, d_{ko} are the flags that the element belongs to the checked part of the basic equipment and the check equipment; $d_{y,1}$ is the sentinel of the effect of the element on the status of the elements of check level 1. If the component equals zero, then the flag given for it is absent.

The definition of ρ consists of executing the following five stages. At the first one is imitated the appearance in the object of a malfunction (failure) [5] due to a problem with the i-th element of the DO, and the status of the elements computed in models M and M^*_i (See blocks 3-11 in Fig. 1). Joint operation of the models (block 10) is executed only from cycle χ of operation of the DO in which imitation of the problem i must begin (block 8).

At the second stage of solving the problem is directly simulated the operation of SK's of the check level (blocks 12, 13). For this, to the MDO are added three types

of check circuit models (MSK): logic, functional, and logic-functional. The first is used if there is a circuit realization of the SK in the DO; the second, if only the digits of the basic equipment and their check method are defined for the SK; the third, if in the DO are allocated digits of the basic Y_{oo} and check Y_{ko} equipment, and they do not have a decoding circuit.

Simulating the operation of the logic MSK is done from a definition of the presence of an error in the Y_{oo}, Y_{ko} digits and SK reaction to the error by the formulas:

$$d(Y_{oo}, Y^*_{oo})_\alpha = (y_{oo,j} \oplus y^*_{oo,j})_\alpha, \; j = \overline{1, n_{oo_\alpha}}; \tag{2}$$

$$d(Y_{ko}, Y^*_{ko})_\alpha = (y_{ko,j} \oplus y^*_{ko,j})_\alpha, \; j = \overline{1, n_{ko_\alpha}}; \tag{3}$$

$$d(y_{ck}, y^*_{ck})_\alpha = (y_{ck} \oplus y^*_{ck})_\alpha, \tag{4}$$

where $d(Y,Y^*)$ is the function of mod 2 summation of like digits SK_α in the models M and M*; $\alpha = \overline{1, n_{ck}}$ is the ordinal number of SK; Y_{oo}, Y_{ko} (Y^*_{oo}, Y^*_{ko}) are the set of digits of the basic and check equipment monitored by CK_α in models M and M*, respectively; y_{ck} (y^*_{ck}) is the output of CK_α in these models; n_{oo}, n_{ko} is the number of digits in Y_{oo}, Y_{ko}, respectively.

If expressions (2)-(4) do not equal zero, then at the outputs of Y^*_{oo}, Y^*_{ko}, y^*_{ck} are errors. In the functional MSK, the state of Y_{ko}, y_{ck} is determined by the MSK by conversion of the states of Y_{oo}, y_{ck} according to the check method algorithm. If $d(Y_{oo}, Y^*_{oo})=1$, the error will be detected. In the logical-functional MSK, determination of the status of Y_{ko} and Y^*_{ko} and formation of the error sentinal in the check and basic equipment is done by analogy with the logical MSK, while the SK error detection sentinal is formed the same as in the functional MSK.

The MSK composition given enables simulation of the work of both the SK's in the DO logic structure and the alternative SK variants not realized in the DO structure.

The d_{ck} result formed by the MSK is independent of the model type, and has the following information structure:

$$d_{ck,\alpha} = \{d_{oo};\; d_{ko};\; d^*_{oo};\; d^*_{ko};\; d_o;\; d_i\}_\alpha \,. \tag{5}$$

Here, all the components are symbols of the binary alphabet (1,0). If the component equals one, then d^*_{oo}, d^*_{ko} are sentinels of the appearance of an error in digits Y^*_{oo}, Y^*_{ko}, checked by CK_α; d_o, d_i are sentinels of detection and correction of error by CK_α. If the component equals zero, then the sentinel given for it is absent.

To reveal errors of the unchecked part of the equipment, we combine all the unchecked DO outputs into one group, which we will conditionally view as an additional check level. To analyze the status of this level's outputs, we add to the MDO one more MSK type, identical to those above in the structure of the result formed and the interconnection with the MDO. The MSK introduced differs in that it computes by formula (2) whether there is an error on the unchecked DO outputs; depending on this, it forms the value of the d^*_{oo} component, always assigning zero to the remaining components of the result formed.

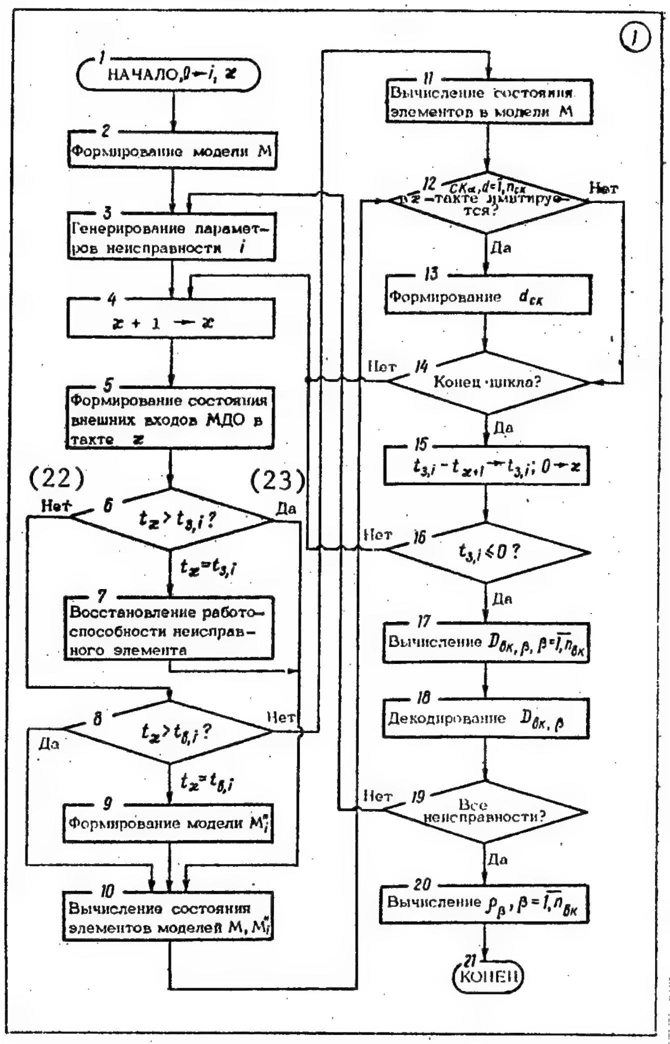


Fig. 1. Structural Scheme of the Problem's Solution

Key:

1. Start
2. Formation of model M
3. Generation of parameters of error i
5. Formation of the status of external MDO inputs in cycle χ
7. Restoration of faulty element's operation
9. Formation of model M_j^*
10. Compute status of elements of models M, M_j^*
11. Compute status of elements in model M
12. In χ -cycle simulated?
13. Formation of d_{ck}
15. End of cycle?
17. Compute
18. Decode
19. All malfunctions?
20. Compute
21. End
22. No
23. Yes

Formation of d_{ck} is done at each reference to MSK while the faulty status of the DO is being simulated resulting from the same malfunction, if the error caused by it is not detected during the preceding reference to MSK. At the third stage is determined the result $D_{vk,\beta}$ of the operation of the check version . $D_{vk,\beta}$ has the following information structure:

$$D_{bk,\beta} = \{D_{oo};\ D_{ko};\ D^*_{oo};\ D^*_{ko};\ D_o;\ D_i\}_{\beta}. \qquad (6)$$

Here, all the components are symbols of the binary alphabet (1,0). The physical meaning of the D_{vk} components corresponds to the d_{ck} components identical by indexes, but each D_{vk} component belongs to a check variant.

In contrast to the d_{ck} components, the D_{vk} components are computed analytically (see Fig. 1, block 17) after completion of simulation of the malfunction i, using d_{ck}, in the variant SK. It is assumed in computation that D_o=1 if the error due to the malfunction is detected by at least one SK of the check variant; D_i=1 if the error is corrected at all check levels. The remaining components of D_{vk} are computed by analogy with D_o = by the value of the d_{ck} components corresponding to them.

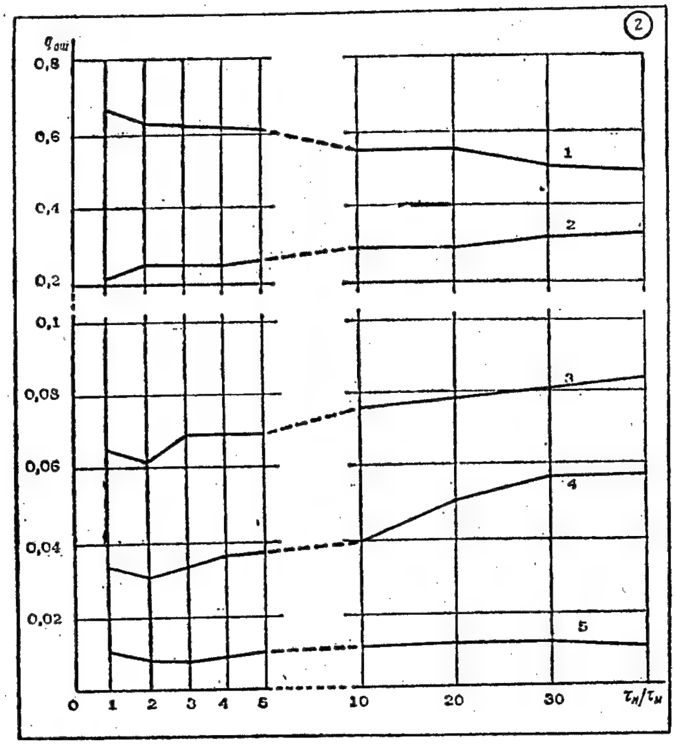
At the fourth solution stage occurs accumulation of data (Table 1) required for determining the probabilities sought. Accumulation of the data consists of decoding (see Fig. 1, block 18) the value of the components of $D_{vk,\beta}$ and adding a unit to the counters of the accumulated data upon fulfillment of the condition of accumulation. The structure of the error at the check levels (for example, error frequency) is characterized by the number of distorted digits of the check level, defined as the number of units in (2). The number of errors of the same frequency is accumulated in the error frequency counters.

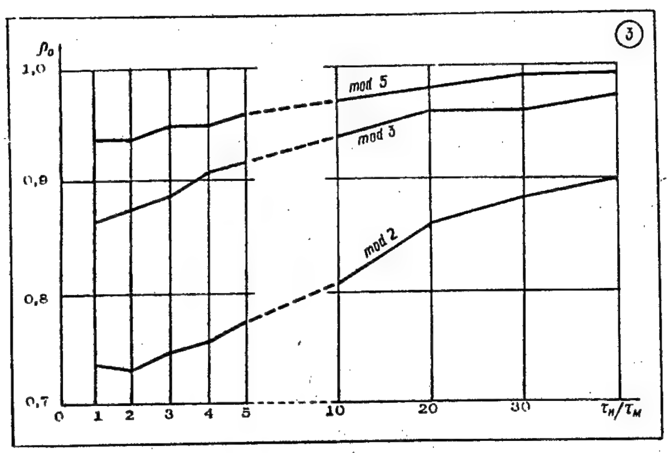
Table 1. Accumulation of Data

Data Accumulated	Accumulation Condition
Number of DO errors	D^*_{oo}=1 or D^*_{ko}=1
Number of errors in the equipment checked by:	
check circuit α	d^*_{oo}=1
check variant β	D^*_{oo}=1
Number of errors detected in the equipment checked by:	
check circuit α	d_{oo}=1 and d_o=1
check variant β	D_{oo}=1 and D_o=1
Number of corrected errors in equipment checked by:	
check circuit α	d_{oo}=1 and d_i=1
check variant β	D_{oo}=1 and D_i=1

Note. For the check equipment, the data accumulated and condition of their accumulation are identical to those given for the checked equipment.

At the fifth solution stage, the efficiency of error correction by the circuits and by check variants and the probability q_{osh} of the error of the determined structure are computed, and the results obtained printed.





Figs. 2 and 3. Effect of the length of malfunction on the probability of errors q_{osh} of varying frequency, and probability of detection ρ_o of errors by SK by modulus.

The particular feature of the practical realization of this method is that it is included as an integral part of the system of probability logic simulation designed for working out the DO logic design. Such a realization of the method has simplified solution of the question of coding raw data on the DO studied, since there are additionally coded only data on the check means and levels, which is less than 10% of the volume of data usually coded when using models to check an object's operating ability. On the other hand, this approach expands the possibilities of logic simulation systems by helping to enhance their efficiency and further development.

Тип неисправности элемента (1)	(2) Вероятность ошибки на выходе ТЭЗ		
	(3) одиночной	(4) двукратной	(5) трехкратной
(0,1)→(1)	0,495÷0,867 / 0,677	0,092÷0,494 / 0,164	0,011÷0,123 / 0,058
(0,1)→(0)	0,262÷0,892 / 0,601	0 072÷0,660 / 0,277	0,004÷0,118 / 0,058
(0,1)→(1,0)	0,196÷0,919 / 0,641	0,045÷0,534 / 0,239	0,006÷0,134 / 0,065

Table 2. Dependence of Error Probability on Component Malfunction Type

Key:
1. Type of component malfunction
2. Probability of error at the TEZ [standard replacement component] output
3. Single
4. Double
5. Triple

This method has been used to study the dependence of frequency and probability of detection of an error by SK by modulus due to malfunctions and failures in logic networks based on components of a medium degree of integration. The study is done on models of ten standard replacement components (TEZ) of the YeS1035 computer processor, mainly those of channels and the arithmetic-logic device. The average degree of integration of components in the TEZ is 50 series 500 IC's. Imitation of the SK operation is done by functional MSK's realizing check algorithms by mod 2,3,5, respectively. The error frequency is established at the TEZ outputs. The length of the malfunction τ_n of the components varies discretely in the range of 1 to $40\tau_m$ (τ_m is the time of simulation of TEZ operation at one input combination of signals). The appearance of 2000 malfunctions in the TEZ is simulated for one malfunction length.

The effect of malfunction length on the error probability q_{osh} of varying frequency and the probability of detection ρ_o of errors by SK by modulus are given in Figs. 2 and 3. Table 2 gives the dependence of the probability of errors on the type of component malfunction given a malfunction length equal to τ_m. The numerator here defines the range of change in the error probability; the denominator, the mean value.

This method is also used to determine the probability of detection by hardware error check in the arithmetic-logic unit of the processor of the YeS1035 [6].

Conclusions. The predominating information errors due to failures and malfunctions of equipment are single and double (accounting for 90% of the errors). Single errors resulting from malfunctions amount to 65%; double errors, 25% of all errors. Failures result in an increase in the probability of multiple errors. The probability of single errors due to failures declines by 25% on the average; that of double and triple errors increases by 40 and 30%, respectively.

Errors caused by malfunctions are detected by the SK by mod 2 with a probability of 0.73; by mod 3, with a probability of 0.87; by mod 5, with a probability of 0.93.

The detecting ability of the SK increases as the malfunction length grows. The rate of increase rises with a reduction in the check modulus. The result is a smoothing out of the difference between check moduluses in terms of failure detection probability.

The probability of malfunction detection is less than that of failure detection by SK's by mod 2 by 20-30%; by mod 3, by 5-15%; by mod 5, by 2-10%.

The probabilities of errors of identical frequency given identical types of malfunctions of components in different TEZ's vary over considerable ranges: identical errors up to five times; double, to 12 times; triple, to 30 times.

The special feature of the practical realization of the method is: the variable length of the simulated malfunction; simulation of SK's with a different degree of detail in their structure in the DO; and inclusion of the method in the system for logic simulation for checking DO operating ability.

BIBLIOGRAPHY

1. Ushakova, G.N., "Apparatnyy kontrol' i nadezhnost' spetsializirovannykh EVM" [Hardware Checking and the Reliability of Special-Purpose Computers], Moscow, Sov. radio, 1969, 312 pages.

2. Sellers, F., "Metod obnaruzheniya oshibok v rabote EVM" [The Method of Error Detection in Computer Operation], Moscow, Mir, 1972, 310 pages.

3. "Obnaruzheniye i ispravleniye oshibok v diskretnykh ustroystvakh" [Detection and Correction of Errors in Discrete Devices], edited by V.S. Tolstyakov, Moscow, Sov. radio, 1972, 288 pages.

4. Lipayev, V.V., Kalinichenko, I.M., "The Reliability of Data Processing in ASOU [Automated Systems for Organizational Control]", USiM, 1979, No 1, pp 3-6.

5. Liokumovich, B.Ye., "Probability Simulation of Malfunctions of Discrete Objects", USiM, 1981, No 2, pp 55-59.

6. Liokumovich, B.Ye., Oleynik, A.V., Osipov, B.G., Shklyar, V.B., "A Study of the Efficiency of Checking the Arithmetic-Logic Unit of the Processor of the YeS1035 Computer", VOPROSY RADIOELEKTRONIKI, SER. EVT, 1978, Vol 12, pp 10-17.

9875
CSO: 1863/140

UDC Δ65.012.2:002.5.65.011.56

COMPARATIVE CHARACTERISTICS OF METHODS FOR ORGANIZATION AND TECHNOLOGY OF SOFTWARE DEVELOPMENT FOR MANAGEMENT INFORMATION SYSTEMS

Moscow MEKHANIZATSIYA I AVTOMATIZATSIYA PROIZVODSTVA in Russian No 4, Apr 83 pp 29-32

[Article by A.P. Ivanov, doctor of economic sciences, and N. V. Tychina, engineer]

[Excerpts] The expansion of the field of use of computers and the rapid growth in speed of modern computers are attended by a significant increase in personnel engaged in data processing. Low programmer productivity, despite the use of automation facilities, is causing lengthy schedules for development of management information systems [MIS] and substantially increasing labor's share in system cost. For example, the share of costs accounted for by software development in our country grew from 44 percent in the period 1971-1975 to 55 percent in the period 1976-1979 and is forecasted to reach 75-80 percent by 1985. Under the conditions of limited growth in labor resources while intensive factors of economic growth are becoming decisive, the problems of increasing programmer productivity and raising program quality and reliability are becoming extremely relevant. A special role in solving these problems is played by the factors of organization and technology in developing software for MIS's.

Special attention has recently been paid to methods, procedures and style of programming, known under the general title of "improving the organization and technology of programming." These methods are being developed not to counterbalance facilities for automation of programming on the base of modern program facilities (translators, applications software packages, systems for computer-aided design of efforts, etc.), but to supplement them, enabling reduction in project development schedules and the capability of efficiently managing efforts irrespective of the organizational forms of using computers, and the programming facilities and operating envirnoment employed.

In domestic and foreign practice, there are quite a few methods of organization and technology for software development including the efforts on programming, debugging and documentation. It seems useful to compare them and indicate the fields and conditions for efficient use of them.

It should be noted that these approaches to software development can somewhat lower software efficiency from the viewpoint of use of machine resources. For example, using structured code in programs requires 6 to 10 percent more memory and program execution time may also be increased.

But the considerable sizes of main and external memory and the processor speed in modern computers allow offsetting these shortcomings efficiently. Also, the purpose in using these new methods is to save labor input in the processes of MIS software development and maintenance which is a major national economic problem.

8545
CSO: 1863/160

UDC 621.3.049.75.001.63:681.3.06

DIALOG SYSTEM FOR DESIGNING PRINTED CIRCUIT BOARDS AT AUTOMATED WORK SITE COMPLEX

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83 (manuscript received 30 Mar 83, after revision 17 Aug 82) pp 32-36

[Article by A. Ya. Vol'fenzon, G. P. Demidov, D. Ye. Zapolotskiy, V. I. Peskov and D. M. Shteyman]

[Text] Introduction. On-line interaction between the designer and the computer-aided design system is a necessary element in modern design automation systems, not an auxiliary support service.

This paper describes a dialog system for designing two-sided printed circuit boards (DSPPP), based entirely on the ARM-R. This distinguishes it from several current systems, where as the basic hardware is used either an ARM [automated work site] together with a powerful computer [1,2], or a powerful computer equipped with additional graphic data I/O equipment [3,4]. Some common systems have no on-line dialog at all.

It is quite obvious that from the economic standpoint the use of minicomputers is considerably more profitable than the use of a powerful computer in terms of both cost and operating characteristics. The speed of the SM-4 computer is almost as good as that of medium YeS computer models, while the rather sophisticated peripheral equipment of modern ARM's provides all the preconditions for solving the issue of a dialog mode in the design process. Failure to use a powerful computer results in the necessity of developing and certifying algorithms and programs that, on the one hand, are oriented towards a small working memory, and on the other, are fairly fast. These requirements are somewhat contradictory.

Nevertheless, the possibility of their fulfillment is confirmed by a number of foreign developments [5-7]; in particular, new products of the firms Quest [6] and RACAL-REDAC [7], where printed circuit board design systems including complexes of location and tracing programs have been created based on minicomputers.

Requirements for the DSPPP, and Basic System Characteristics. Of all the general requirements considered in developing this system, the following five groups can be conditionally singled out: operational; functional; requirements on production of design documentation; hardware; and software.

As noted in [2], formulation of the operating requirements presents a certain problem. Without going into possible solutions to this problem in detail, we shall note that the basic focus here is on the principle of system "communicability", formulated in [8].

The main operating requirement is a dialog mode of system operation at all design stages of the printed circuit board (data preparation, arrangement of elements on the board field, tracing circuits, technological control, production of design documentation and control punch tapes), providing the designer with the possibility of on-line work with graphic information. Another important requirement is simplicity of operation with the DSPPP for a user unfamiliar with basic programming. This is achieved by terminologically similar, problem-oriented languages describing the input information and directive procedures.

Functional Requirements.Those placed on the DSPPP depend on the quality criteria and technological constraints imposed on the objects designed. They are essentially requirements on functional algorithms. Such requirements include the following: satisfaction of the criterion of a minimum total number of connections in placement of the components on the printed circuit field; execution of circuit tracing taking into consideration criteria of the minimum total length of circuits and minimum junction holes; and ensuring technological control of bottlenecks.

An important functional requirement consists of the speed constraints required for achieving the dialog design mode. In particular, the requirement of ensuring a mean time for obtaining one trace of about 5 seconds is imposed on tracing programs in the DSPPP.

Requirements on Producing Design Documentation. These are defined by the standards in effect.

Technical Requirements. These assume the following version of system operation at the ARM-R complex: SM-3 (SM-4) processor; working memory, at least 28K words; IZOT-1370 magnetic disk memory; EPG-400 (EPG-SM) graphic display unit; AP-7252 (YeS7052) plotter; alphanumeric display unit; mosaic-serial printer; and papertape I/O units. The basic technological equipment consists of M-2001 (M-2004, M-2005) coordinatographs, CARTIMAT, and "Shmol'" automatic drilling machines.

The Program Requirements. These basically involve SAPR [automated design system] universality, its capability for quick adjustment given a possible switch to a new computer model or another operating system. They must also ensure simple connection of new program modules to the system. The DSPPP software is based on the modular principle, written in FORTRAN with minimal use of a macroassembler, and is under RAFOS control.

DSPPP Structure. The system is represented by a set of subsystems, operating under monitor control and interconnected with each other. They include: language, functional, dialog, I/O, and librarian. The structure of such a system is given in the figure.

Language Subsystem. The SAPR software must have developed resources organizing user communication with the system in the terms and concepts of the field in which the development is taking place; i.e., in a language close to the user's professional

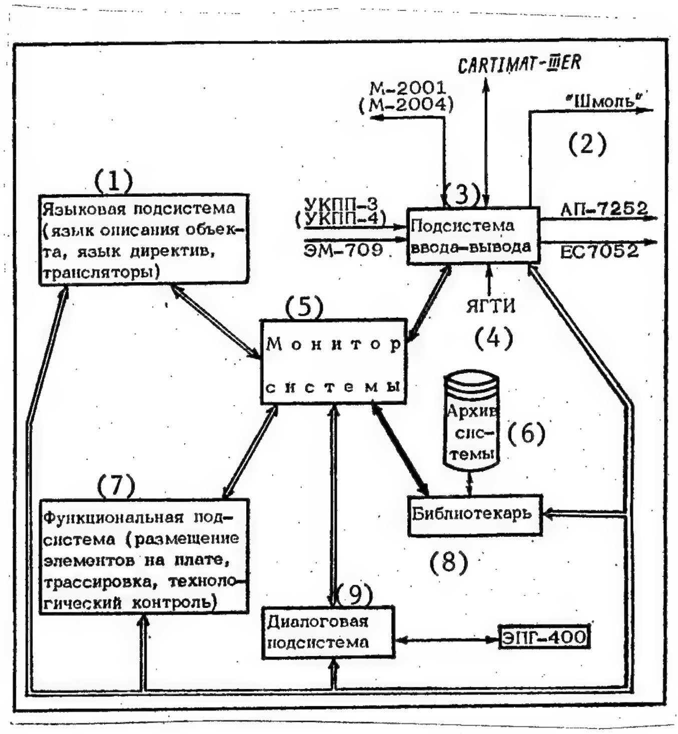


Fig. 1. Structure of the DSPPP

Key:

1. Language subsystem (object description language, directive language, translators)
2. "Shmol'"
3. I/O subsystem
4. YaGTI [a language for representation of graphic information]
5. System monitor
6. System files
7. Functional subsystem (location of components on the printed board, tracing, technological control)
8. Librarian
9. Dialog subsystem

language. The totality of such resources form a separate subsystem in the DSPPP, consisting of languages of interaction with the system and translators from these languages. At the present time, a language describing circuit elements, circuits, tracing forbidden regions, power supply buses, etc., is used for loading data on the object being designed. A fragment of the raw data's description is given below.

```
* EXAMPLE OF MICROCIRCUIT (COMPONENT) DESCRIPTION
MK (3; 1; 10,30; 4)
 ;3--NUMBER IN THE SERIES
 ;1--TYPE
 ;10,30--ATTACHMENT COORDINATES
 ;(OPTIONAL PARAMETER DURING
 ;AUTOMATIC PLACEMENT)
```

```
 ;4--ORIENTATION (OPTIONAL PARAMETER)

*EXAMPLE OF DESCRIPTION OF AN INDIVIDUAL CONTACT AREA
KP (65; 99; 20,10)
 ;65--NUMBER IN THE SERIES
 ;99--TYPE, CORRESPONDING TO A DIAMETER OF 1.44 MM
 ;20,10--COORDINATES

*EXAMPLE OF DESCRIPTION OF THE FORBIDDEN ZONE
Z(1;10,20;10,26;15,26;15,20;10,20)
 ;1--CRITERION OF ZONE LOCATION ON LAYER 1
 ;FURTHER--COORDINATES OF BEND POINTS
 ;LINES DELINEATING THE ZONE

*EXAMPLE OF DESCRIPTION OF THE POWER SUPPLY BUSES
Sh(3;1.25;10,21;15,21;15,20;24,20)
 ;3--CRITERION OF LOCATION OF THE POWER SUPPLY BUS
 ;ON BOTH SIDES OF THE BOARD
 ;1.25--BUS WIDTH IN MM
 ;FURTHER--COORDINATES OF THE BEND POINTS
 ;LINES DESCRIBING THE BUS

*EXAMPLE OF DESCRIPTIONS OF CIRCUITS Ts (N1, B1; N2, B2; N3, B3;...)

*N1--NUMBERS OF MICROCIRCUITS, B1--NUMBERS OF OUT-
 ;PUTS OF Ts (3,9;3,10;2,5;4,5)
 ;CIRCUIT DESCRIBED CONNECTING
 ;9 AND 10 OUTPUTS OF MICROCIRCUIT WITH NUMBER 3,
 ;5TH OUTPUT OF MICROCIRCUIT WITH NUMBER 2 AND
 ;5TH OUTPUT OF MICROCIRCUIT WITH NUMBER 4
```

The directive language controls the process of automated design. In the directive information, one can single out operators of I/O control, variation in system parameters, activation of various modules of the functional subsystem, etc. The directive information is interpreted by the dialog system monitor, and usually generates a call of a certain overlay section realizing the required function.

Functional Subsystem. This usually includes two program components: programs of placement of the radio components on the printed circuit board; programs ensuring a trace between a specified set of radio component contacts (tracing programs).

The layout algorithm in the DSPPP uses an electric analogy to simulate the components of the printed circuit board and their connections [9]. The mathematical model of the problem is represented as a graph G(X,U), whose vertices X are both the modules and the connections between them (circuits). The branches of the graph (U) are a pair representing the modulus-circuit connection. The conductivity of each graph branch is compared, and a matrix of conductivity, usually very thin and isomorphous to the graph, is constructed. To speed up solution of the systems of linear equations represented by the thin matrices, an algorithm is used that makes it possible to find the solution without changing the structure of the initial matrix. It is based on a modified Kachmarzh method [10], defining the algorithm's speed values.

The layout programs are certified during the design of boards made on one-size components. The results of machine and manual placement are compared by the criterion of total length of all connections. The result averages 10% better for placement using DSPPP programs than with manual placement.

The tracing algorithm developed for the DSPPP can be placed in the class of main line algorithms [11]. It is based on finding the connected subgraph that includes a specified set of points on the full graph of unoccupied intervals. After the subgraph defined by the given circuit is isolated, it is excluded from the full graph.

The speed of the tracing algorithm is defined by the system of lists mapping the graph of unoccupied intervals. Operations for working with the lists have been realized.

When developing the tracing algorithm, the basic attention is focused on its speed values. The algorithm developed and its program realization ensure the fulfillment of conditions under which the mean time for performing the trace is less than 5 seconds, which makes it possible for a person to interact on-line with the tracing program.

Dialog Subsystem. This subsystem is singled out as a separate, independent part, since it is the main element ensuring inclusion of a person in the CAD process at any stage. The basic form of such a dialog is the graphic dialog; i.e., information is displayed on the graphic display's screen, followed by its correction and further use by the system's programs. The dialog subsystem consists of a complex of programs ensuring the following operations: input into the system of parameters controlling the design process; indication of the system's current status; display of graphic objects on the graphic display's screen; and correction of graphic objects using functional and alphanumeric keyboards and a light pen.

The dialog subsystem is connected with almost all the other subsystems. Its core is a complex of programs for editing graphic objects. Besides functions of display on the graphic display's screen, this complex's capabilities include: isolation of the graphic object using the light pen, erasing the object, or shifting it in any of eight directions that are a multiple of 45°; adding new graphic objects (drawing mode), which can be: line (conductor), point (contact spot, junction hole), text (marking); the graphic object has certain attributes (for example, line type or width, junction hole diameter, text height, etc.), the suppression mode is possible; means for unblocking a complex object, in other words, breaking it down into a series of components ("breaking off" a section, dividing a broken line into sections, etc.) for subsequent correction of individual parts of the graphic object; means for grouping several simple objects into one complex one for further work with it as a single object; and means for working with graphic objects in the mode of a so-called four-fold window, in other words, merging objects located in any of the four quadrants relative to an arbitrary point of the printed circuit board. Graphic editing in this mode is a convenient means when designing complex, saturated boards.

All the basic instructions most frequently used in the process of correcting graphic objects are designed for the EPG-400 (EPG-SM) function keyboard. Individual instructions, relating to I/O or modification of parameters of objects and the

system as a whole, and requiring input of certain text information, are designed for an alphanumeric keyboard.

The complex of graphic editing programs is written in FORTRAN with minimum use of a macroassembler. It is realized as a separate overlay section of the dialog subsystem, and can be used in other problems where correction of graphic information is required (for instance, in automated manufacture of photo templates).

The Graphic Information I/O Subsystem. This subsystem provides connection of the dialog subsystem with graphic information I/O devices at various design stages. The connection is realized by converting information from the internal representation of data of the dialog subsystem into codes of the executive devices (output); or, on the other hand, codes of the executive devices into the internal representation (input). By executive devices are here meant both those for coding and representing graphic information, and other design systems having their own language for representing graphic information (for example, YaGTI [12]).

The basic requirement imposed on the I/O subsystem is simplicity of connecting a new device to the spectrum of devices served by the subsystem. Its basis is the principle of standardized connection of a new device with the dialog subsystem. This does not mean standardization of any graphic information representation language and development of programs for recording from this language for the given device (as is done in the GRIF system [1]), but standardization of program inputs and outputs of the dialog subsystem (as done to some extent in the GRAFOR system [13]). In the I/O subsystem under consideration, input is realized from a certain virtual input device to the dialog subsystem, and output from the dialog subsystem to a certain virtual output device. The task of connecting the new device thus consists of displaying the virtual device for one or another actual one. Each I/O program is realized as an overlay section controlled by the dialog subsystem's monitor.

The I/O subsystem currently handles: input of graphic information in codes of the coordinatographs M-2001 (M-2004), CARTIMAT-IIIER; input of graphic information in YaGTI; input of graphic information prepared on coding devices UKPP-3, UKPP-4 (a further improvement of the UKPP-2 [14]) and EM-709; representation of graphic information on AP-7252 and YeS7052 (plotters) units; production of a control papertape for coordinatographs M-2001 (M-2004), CARTIMAT-IIIER; and production of a control papertape for the "Shmol'" automatic drilling unit.

Librarian. The basic function of this subsystem is to support the files, consisting of descriptions in the internal structure of the data of standard elements, intermediate or final design results, descriptions of raw data, etc. The system's file is on magnetic disk, and can be corrected using control directives interpreted by the system monitor. Functions of adding new elements to the file, reading from the file and purging a data element from the file are currently realized. An archive element is identified by its name.

Conclusion. There is presently in use a DSPPP version based on the ARM-R hardware complex with a 28K word working memory and the equipment described in the technical requirements for the DSPPP. The system is used in designing digital printed circuit boards. Placement of elements on a printed circuit board with dimensions of 170x200 mm (discrete 1.25 mm) with 50-60 microcircuits and average of about 200

circuits lasts 1.5-2 minutes. Tracing such a board in the automatic mode takes 3-5 minutes and provides an average layout of 95% of the connections. The file obtained from the functional subsystem's operation can be outputted to the screen of the graphic display (EPG-400), edited using graphic correction programs, drawn (AP-7252), written into the file, or converted and outputted on punched tape for technological equipment (M-2001, "Shmol'").

A second version of the DSPPP is being readied for introduction, providing the capability for interruption of the automatic tracing mode with output to dialog after a specified number of traces made.

BIBLIOGRAPHY

1. Yelshin, Yu.M., "Design of Printed Circuit Boards Using the ARM-R", OBMEN OPYTOM V RADIOPROMYSHLENNOSTI, 1979, No 10, pp 109.

2. Gerasimenko, Ye.P., Kot, V.I., Landau, I.Ya., Somkin, V.M., "Avtomatizatsiya proyektirovaniya pechatnykh blokov s modulyami proizvol'noy formy" [Automated Design of Printed Blocks with Modules of Arbitrary Shape], Moscow, Mashinostroyeniye, 1979, 168 pages.

3. Bakhtin, B.I., "Avtomatizatsiya v proyektirovaniya i proizvodstve pechatnykh plat radioelektronnoy apparatury" [Automation in Design and Production of Printed Circuit Boards of Electronic Equipment], Leningrad, Energiya, 1979, 120 pages.

4. Orlovskiy, G.V., Kazenova, I.I., Kval'vasser, Yu.G., et. al., "The Architecture for a System of Automated Design REA", OBMEN OPYTOM V RADIOPROMYSHLENNOSTI, 1975, No 7, pp 11-14.

5. Rosa, R.C., Lucio, T.P., "Programs for Two-Sided Printed Circuit Board Design", COMPUTER-AIDED DESIGN, 1979, Vol 11, No 5, pp 297-303.

6. Plus fuer Leiterplatten, Entwurfe, ELEKTRON. ZTG., 1980, No 20, p 31.

7. Khirau Eydzi, "The REDAC-MINI System, Operating in a Dialog Mode", DENSI GIDZYUTSU, 1980, Vol 22, No 6, pp 55-57.

8. Zapolotskiy, D.Ye., Karpenko, S.N., Kuzin, S.G., et. al., "Construction Principles and Architecture of a Package of Application Programs", USiM, 1979, No 1, pp 8-14.

9. Petrenko, A.I., Tetel'baum, A.Ya., "Formal'noye konstruirovaniye elektronno-vychislitel'noy apparatury" [Formal Design of Computer Equipment], Moscow, Sov. radio, 1978, 256 pages.

10. Zapolotskiy, D.Ye., Mokichev, N.M., "A Modified Kachmarzh Method for Solving Systems of High-Order Linear Algebraic Equations", PRIKLADNYYE PROBLEMY PROCHNOSTI I PLASTICHNOSTI, 1976, No 3, pp 31-32.

11. Selyutin, V.A., "Mashinnoye proyektirovaniye elektronnykh ustroystv" [Machine Design of Electronic Devices], Moscow, Sov. radio, 1977, 384 pages.

12. Vermishev, Yu.Kh., Yelshin, Yu.M., Bibikov, V.F., Lomakina, Z.I., "On the Structure of Graphic and Textual Information in Automated Design Systems", OBMEN OPYTOM V RADIOPROMYSHLENNOSTI, 1977, No 2, pp 22-23.

13. Bayakovskiy, Yu.M., Lazutin, Yu.M., Mikhaylova, T.N., Mishakova, S.P., "GRAFOR: kompleks graficheskikh programm na FORTRANe. Struktura i osnovnyye printsipy" [GRAFOR: A Complex of Graphic Programs in FORTRAN. Structure and Basic Principles], Moscow, 1975, 60 pages (IPM AN SSSR, preprint No 90).

14. Gusev, S.N., Gribunin, A.V., Sazanov, V.V., "A Device for Encoding Printed Circuit Board Blueprints", OBMEN OPYTOM V RADIOPROMYSHLENNOSTI, 1975, No 6, pp 155-156.

9875
CSO: 1863/140

UDC 519.685.4

SUPPORT OF FUNCTIONS OF OPERATING SYSTEM OF 'EL'BRUS' MULTIPROCESSOR COMPUTER COMPLEX IN TEMP INSTRUMENTAL COMPLEX

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83 (manuscript received 15 Jun 82, after revision 22 Dec 82) pp 49-52

[Article by A.A. Gutman, "Support of Functions of the Operating System of the "El'brus" MVK [multiprocessor computer complex] in the TEMP Instrumental Complex"]

[Text] Introduction. The development of new computers and their software (PO) frequently uses instrumental complexes of system programs simulating a special-purpose (new) computer on the instrumental (old) one. The instrumental complexes enable the new computer's PO to be created in parallel with its hardware part. For example, the instrumental complex for the IBM/7010 was used during creation of the IBM/360 and its operating system OC/360 [1].

The usefulness of instrumental complexes grows sharply if they can be used to create and debug not only operating systems, but other software, as well as production programs for a special-purpose computer. However, such programs generally use different OS functions provided to the programs (for example, memory, file and process control, to name some of the most important functions).

In realizing the "El'brus" MVK [2,3] on the BESM-6, the STRELA complex [4] was created. Its operating experience has confirmed the need to provide OS functions in instrumental complexes. Therefore, in the instrumental complex replacing it, the TEMP [5,6], the problem of ensuring MVK OS functions was solved first by one method, then another. This article deals with this subject.

In addition to the monitor, editor and other instrumental and service resources, the TEMP complex [6] contains two components simulating the MVK "El'brus": a translator from the autocode [7] (the autocode is the basic programming language in the MVK [8]) to the file of the MVK object code; and the MVK "El'brus" interpretor [9], effecting (interpreting) the file of the MVK object code. The complex includes also translators from ALGOL-60 and the standard FORTRAN [10], developed themselves on the TEMP complex and included in the standard MVK "El'brus" software. The MVK OS functions are handled in the TEMP complex by the interpreter, one of whose parts is the OS model. Moreover, a so-called large interpretor is being developed, in which the MVK OS functions are handled differently: the programs are interpreted under control of the real MVK OS.

This article examines an OS model in the interpreter of the TEMP complex, the main problems and principles of its realization. The development and operation of the OS model is analyzed, from which the need to create a large interpreter becomes clear.

Basic Problems in Realizing the OS Model. A posteriori, the main problems in simulating OS functions in the interpreter are: the technical difficulty of programming special-purpose computer functions in the language of the instrumental computer; the search for optimal representations in the model of various OS mechanisms; realization under conditions of incomplete external specifications; and the fast growth in user requests.

The second problem is the usual task of designing a simulating system (see, eg, [11]). The others should be examined in greater detail.

Interaction with the Instruction Interpreter. The following decisions in interpreter design enabled a subsequent considerable reduction in the difficulty of programming various MVK effects in the realization language of the TEMP complex YaRMO [12,13]. The basic portion of the interpretor, the instruction interpreter, provides the OS model with access to several procedures enabling programming of all the necessary actions in MVK terms at a rather high level. For example, there are two procedures for reading and writing into the MVK's mathematical memory: READ BY MATHEMATICAL ADDRESS and WRITE BY MATHEMATICAL ADDRESS. The first procedure parameter is the address; the second specifies where the MVK value should be read to (where it should be written from) in the instrumental computer.

Access to all simulated MVK registers and the main operations with them, such as those with the MVK stack, are handled similarly. Finally, the instruction interpreter provides the OS model with the EXECUTE INSTRUCTION procedure, which executes any instruction (specified by parameter) of the MVK's CPU.

Method and Basic Criteria of Realization. The problem of realization under conditions of incomplete external specifications arises because the programs and software created on the instrumental complex must sometimes be ready when the special-purpose computer is delivered to the user. This means in practice that the complex is developed in parallel with the hardware portion, and with the special-purpose computer's OS. The TEMP complex was thus created in parallel with the "El'brus-1" MVK and MVK OS.

Work under conditions of increasingly accurate (and sometimes changing) external specifications led to a search for a special realization method, initially used intuitively, then consciously. It can be called a clearer definition of the known method of realizing operating systems by layers [14], in which the following criterion is used to divide the system into layers: everything recognized in the specifications as being established is classified in the internal layer. Floating and absent portions of the specifications go into the higher (external) layers of the system. Uncertainty of the external layers therefore does not prevent realization of the system's lower layers.

It should be pointed out that the most important characteristics of the program product (besides accuracy of simulation) are the efficiency of realization and flexibility of the model. The method of realization chosen enables these criteria to be met.

Selection of Simulation Accuracy. Since the accuracy of the simulation determines the range of possible applications of the model and the basic production characteristics of the system, it is largely responsible for determining realization of the simulating systems. Its correct selection should contain as accurate a description as possible of the system's future use.

In our case, the OS model must ensure execution of OS functions for only unprivileged programs. Moreover, programs created by the translator from the autocode are generally used, although translators debugged on the complex can also send the programs they have generated to be interpreted.

The absence of privileged programs of the special-purpose computer makes it possible in some cases to substantially simplify MVK OS simulation without loss of compatibility. For instance, representation of a mathematical memory for physical and secondary ones is a rather complex problem in the MVK OS. In the interpreter, it is statically divided into sections, corresponding to the total memory of the system, the stack memory, the dynamically ordered memory. Work with each type of memory is done in a different fashion.

The second situation in OS model use named above (the predominance of autocode programs) sometimes makes it possible to confine oneself to compatibility at the autocode input language level in place of compatibility at the code level. This requires a (slight) change in the generation unit in the translators at the output to the MVK.

The basic principle of OS model formulation can be basically formulated: lower simulation accuracy within the limits of use requirements; i.e., simplify simulation of OS functions as much as possible.

Description of the Model of the MVK OS in the Interpreter. The model of the MVK "El'brus" OS in the interpreter of the TEMP complex (as part of the program of the interpreter in the realization language YaRMO [12,13]) consists of several modules. Each module represents a certain realization layer, in accordance with the method of realization by layers described above.

Concept of the OS Model. Among the first accurately defined portions of the MVK OS specifications were the method of processing interruptions as a hardware start of certain OS procedures, and the method of program access to OS functions via a certain memory region, where the labels of the OS procedures executing these functions are located. (In the MVK "El'brus", each word also has a hardware-monitored field ("tag"), specifying the type of value. For example, the label of the procedure is the value of a certain type required to start the procedure [3]). The task thus arises of finding the start mechanism during interpretation of the MVK code of procedures simulating (in the instrumental computer code) the OS procedures.

The concept, or basic idea of realization of the OS model, represents the following solutions to this problem: reading of the procedure label and starting of the OS procedure are done using the same executed instructions and in the same order as in the original system, but the label of the OS procedure has a value type added in the interpreter: the label of the BESM-6 procedure. A procedure start by this label is interpreted as execution of one of a set of procedures simulating the OS, identified by a number contained in the label of the BESM-6 procedure.

This concept is provided to a program using the OS procedures regardless of whether these procedures are simulated or interpreted. Just as importantly, the concept is transparent and easy to realize.

The start mechanism of simulating procedures is realized as the most internal module (first layer) of the OS model.

Model of Interruption Processing. Given the appearance of an interruption in the "El'brus", there is a hardware start of the procedure whose label is chosen from a defined array by the interruption number. The interruption module (the second layer of the OS model) at the beginning of operation fills the array of labels of procedures of reaction to interruptions with the required BESM-6 labels, after which the interruptions are processed as in the original system.

One of the simulating procedures of the OS (REDEFINE) enables redefinition of the reaction to the interruptions; i.e., a specified user procedure is started in place of the standard one. For the redefinition, the label of this procedure is written into the above-mentioned array of labels in place of the standard label of the BESM-6 procedure.

A Review of OS Model Filling. Procedures of the OS model corresponding to the labels of the BESM-6 procedures can be divided into four classes: procedures of interruption processing (in the "El'brus", besides emergency termination of a task, the interrupt mechanism is used to execute several OS functions, such as for dynamic order of memory. These functions are simulated by procedures of reactions to interruptions of the OS model); the procedures of OS interface themselves; procedures of simulating the MVK software (labels of the corresponding procedures in the original system are accessible through other mechanisms; for example, by a dynamic acquaintance with the procedure); and procedures not (yet) having an analog in the "El'brus" software.

The procedures of the first three classes basically simulate the most important functions of the low-level OS, usually started by the interrupt mechanism. For OS functions of a higher level, another approach, leading to the appearance of a fourth class of procedures, is advisable. It is discussed below.

The Principle of Interface Modularization. Assume a certain class of programs must use a common resource. According to the ideas of abstract data types [15], access to such a resource can be provided as to a package of functions. Compatibility in this case is provided by the fact that the package is also realized on the special-purpose computer. Such a package can easily be built into it by using the common mechanism of the first layer of the OS model.

Applying this principle to "El'brus" translators developed at the TEMP complex has made it possible to specify and realize in the interpreter packages of ten procedures, providing generation of the object code file, as well as the INPUT STRING procedure for reading the input string. The procedures are realized on the YaRMO, then (during transfer of the translators to the MVK) again realized on the autocode using the MVK OS file system.

Based on a similar analysis of other requests, a package of graphic procedures has been built into the interpreter, providing development and debugging of graphic programs for the "El'brus" MVK, and a package of dialog programming procedures, providing creation of dialog programs in the EL'BRUS autocode. Graphic files have not yet been realized in the MVK OS. This example shows that the principle of interface modularization sometimes makes it possible to omit developing special-purpose computer software in the OS model by using corresponding instrumental computer software resources in place of still nonexistent special-purpose computer OS resources.

Apparently, the basic constraint on the use of this principle is conceptual. It is advisable to use it when a package function can be defined such that requests from a wide range of users can be satisfied, and simple realization on the special-purpose computer and acceptable generation costs are possible, despite the dual realization (on the instrumental and special-purpose computers).

To meet these requirements, there must be a sufficiently thorough and unified understanding of the essence of the functions to be provided. Unfortunately, this is not the case with many OS functions, in contrast to code generation and input string reading.

Basic Results from Use of the OS Model. The experience from four years (1979-82) of operating the TEMP complex, including the MVK OS model (in the MVK "El'brus" interpreter), enables the following to be formulated as the main operating results.

Utility. The OS model allows a considerable expansion of the class of problems effectively developed on the instrumental complex, and a substantial rise in the level of their preparedness when going to the "El'brus" MVK, in comparison with the STRELA complex. This is confirmed by the completed experience of developing translaters from ALGOL-60, FORTRAN and PASCAL, on-going development of translators from COBOL, FORTRAN-IV and an expanded ALGOL-60, and development of production programs.

Efficiency and Development Costs. Simulation of OS functions has almost no impact on the efficiency of the interpreter (speed, required instrumental computer resources) and the complex as a whole. However, interpreter development costs grew substantially, along with the interpreter volume, compared with the STRELA complex's interpreter.

Operating Problems. The main problem in operation is that of simulating a developing system, which certainly includes the "El'brus" MVK OS. Growing capabilities of the system simulated and the increasing demands of users result in a very fast increase in the demands on the volume of functions simulated. The interface modularization principle described above only partially solves the problem.

The Large Interpreter of the MVK "El'brus". The initial plan for developing the TEMP complex provided for intensive filling of the OS model. It was proposed to simulate work with processes, a system of files, etc. However, it turned out that simulating such a complex and fast-developing system as the "El'brus" MVK OS requires excessively high labor costs. The speed of realization was also unsatisfactory; it is one of the main criteria in OS model development.

The unfeasibility of modeling the MVK OS in the required volume made it necessary to use another realization idea: to not simulate, but interpret the functions of the operating system in the original design. For the MVK OS, it is advisable to interpret the entire operating system as a whole.

Some Realization Aspects. The concept of a large interpreter requires an expansion of the "El'brus-1" MVK model, which necessitates simulation of the physical memory and hardware support of the MVK mathematical memory, the I/O processor and peripheral objects connected to it (magnetic drum, magnetic disk, alphanumeric printer, terminals, etc.), a privileged operating mode of the MVK CPU, and MVK console.

In addition, resetting and restoring the MVK status to terminate and continue the count were necessary, as initialization of the MVK OS in the interpretation mode takes up considerable BESM-6 time. Therefore, the large interpreter's operation begins with the status at the end of OS initialization.

Large Interpreter Capabilities. The radical nature of the proposed concept makes it possible not only to achieve the objective (provide OS functions) at maximum accuracy, but also to obtain several other useful capabilities.

For instance, the large interpreter provides use of the original "El'brus" MVK software (it is advisable above all to use fast-operating components, such as an editor, etc.). Another possibility is debugging expansions and modifications of the MVK OS itself, which is important for certain users of the TEMP complex.

Overall, it can be said that the large interpreter provides complete simulation of the "El'brus" MVK with its software, at an accuracy to the resource characteristics (speed, memory volume).

Conclusion. The requirements of the integral method of computer development and development on instrumental complexes of software and production programs result in the task of providing OS functions of a special-purpose computer, which is solved in the "El'brus" MVK interpreter of the TEMP complex by the simulation method.

The OS model included in the interpreter has considerably enhanced the usefulness of the complex, allowing an expansion of the class of programs generated on it and raising the degree of their readiness during transfer to the "El'brus" MVK. This is confirmed by four years of operation of the TEMP complex at several dozen organizations.

Besides the author, N.V. Golovleva, N.A. Ozols, V.V. Zabotin and E.A. Fel'dman worked on the interpreter, under the direction of S.Yu. Dederer. Besides the author, A.Ye. Sukhov, Ye.B. Bondarenko, L.A. Strikun and S.V. Morozov are working on the large interpreter. The author is deeply grateful to them all, and to V.A. Markov, who reviewed the paper.

BIBLIOGRAPHY

1. Bruks, F.P., "Kak proyektiruyutsya i sozdayutsya programmnye kompleksy" [How Program Complexes are Designed and Built], Moscow, Nauka, 1979, 152 pages.

2. Burtsev, V.S., "Printsipy postroyeniya mnogoprotsessornykh vychislitel'nykh kompleksov "El'brus"" [Construction Principles of "El'brus" Multiprocessor Computer Complexes], Moscow, 1977, 53 pages (ITM and VT AN SSSR, preprint No 1).

3. Babayan, B.A., Sakhin, Yu.Kh., "The AEl'brus" System", PROGRAMMIROVANIYE, 1980, No 6, pp 72-86.

4. Dederer, S.Yu., Markov, V.A., Nikolayev, V.A., et al., "A Software Complex of an Instrumental Computer", in "Sistemnoye teoreticheskoye programmirovaniye" [System Theoretical Programming], Kishinev, 1974, Vol 1, p 76.

5. Golovleva, N.V., Gutman, A.A., Dederer, S.Yu., et al., "An Instrumental Complex For Software Development", in "Vsesoyuznaya konferentsiya po metodam translyatsii: Tez. dokl." [All-Union Conference on Translation Methods: Papers], Novosibirsk, 1981, pp 70-71.

6. Gutman, A.A., Markov, V.A., "The TEMP Instrumental Complex for Simulating the "El'brus" MVK on the BESM-6", USiM, 1982, No 2, pp 95-97.

7. Lesovaya, I.F., Polivanov, V.N., Shishova, N.A., et al, "A Translator from the "El'brus" MVK Autocode: Model and Realization", in "Avtomatizatsiya proizvodstva paketov prikladnykh programm: Tez. dokl." [Automating Production of Packages of Applied Programs: Papers], Tallin, 1980, pp 76-78.

8. Pentkovskiy, V.M., "Avtokod EL'BRUS" [The EL'BRUS Autocode], Moscow, Nauka, 1982, 350 pages.

9. Dederer, S.Yu., "Tekhnologiya sozdaniya proizvodstvennykh interpretatorov bol'shikh vychislitel'nykh kompleksov: Avtoref. dis. ... cand. fiz.-mat. nauk." [The Technology of Creating Production Interpreters of Large Computer Complexes. Authors' Summaries, Candidates of Physical-Mathematical Sciences], Novosibirsk, VTs SO AN SSSR, 1982, 23 pages.

10. Dyatlova, L.G., Kozhukhina, S.K., Lesovaya, I.F., et al, "Realizatsiya ALGOLa-60 i FORTRANa na MVK "El'brus"" [Realization of ALGOL-60 and FORTRAN on the "El'brus" MVK], Moscow, 1978, 15 pages (ITM and VT AN SSSR, preprint No 21).

11. "What is Modelling and Simulation Methodology?", ed. B.P. Zeigler, in Methodology in Systems Modelling and Simulation, Amsterdam, 1979, pp 11-15.

12. Gololobov, V.I., Cheblakov, B.G., Chinin, G.D., "Opisaniye yazyka YaRMO. Yadro yazyka" [Description of the YaRMO Language. Nucleus of the Language], Novosibirsk, 1980, 43 pages (VTs SO AN SSSR, preprint No 247).

13. Gololobov, V.I., Cheblakov, B.G., Chinin, G.D., "Opisaniye yazyka YaRMO. Macrosredstva" [Description of the YaRMO Language. Macroresources], Novosibirsk, 1980, 36 pages (VTs SO AN SSSR, preprint No 248).

14. Dijkstra, E.W., "The Structure of the Multiprogramming System", Commun. ACM, 1968, Vol 11, No 5, p 341.

15. Parnas, D.L., "On the Criteria to be Used in Decomposing Systems into Modules, Commun. ACM, 1972, Vol 15, No 12, pp 1053-1058.

9875
CSO: 1863/140

UDC 681.3.06/94

METHOD OF ORGANIZING DATA FOR SIMULATION OF VIRTUAL EXTERNAL MEMORY OF YeS COMPUTERS AND CREATION OF LARGE-VOLUME DATA BASES

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83 (manuscript received 20 Apr 81, after revision 25 Nov 82) pp 74-79

[Article by L.I. Semchenkov, "A Method of Organizing Data for Simulation of Virtual External Memory of YeS Computers and Creation of Large-Volume Data Bases"]

[Text] Introduction. Solving most problems on the computer requires using external memory devices, of which the most common are magnetic disk and magnetic tape storage. In doing so, the user encounters an increasing shortage of external memory, even in the YeS operating system, which has resources for automatic distribution of free sections on external memory unit volumes.

The main causes for such a shortage seem to us to be the following.

1. First, difficulties in a priori calculation of actual needs for external memory for data sets of a concrete task, with the result that the user distributes the external memory "to the maximum", taking more memory than required from the operating system. Such memory regions, distributed for data sets (using the DD operator and SPACE parameter) frequently remain sparsely filled, but still inaccessible for other data sets.

2. Second, organizational difficulties in monitoring the advisability of allocating external memory for data sets, with the result that disk packs are filled with temporary or file data. Such control in some computer centers requires creating groups of programmers who check the distribution of external memory, reduce user libraries, copy data sets, purge unneeded library sets or sections, etc.

The question of optimum external memory distribution thus remains unresolved. Its urgency rises with an increase in the number of users, which is characteristic of all computer centers, and particularly of collective-use ones.

In addition, questions of automating the distribution of external memory have not been adequately resolved. Distribution of YeS OS memory does not always satisfy the user; for example, he can request 20 cylinders as a continuous section, but his request will be denied if such a section is absent, even though the total size of the free memory exceeds 20 cylinders.

The absence of software controlling memory distribution on magnetic tapes [1] cannot be considered normal. This is widely used in systems processing large volumes of data, due to the insufficient volume of the magnetic disk storage; for example, in the automatic control systems for industrial enterprises, where about 10^9 bytes of information must be stored to create and maintain a complete information model of production [2].

Questions of program realization of a unified (virtual), multi-level external memory are posed in [3,4], but their practical realizations are only applicable for concrete computer configurations, a certain external memory unit set, and a finite memory volume. Automation of the distribution of the entire computer external memory, regardless of the type and number of media, is thus an urgent issue. However, there is no present technique for solving this problem.

The urgency of the problem increases considerably with development of VTsKP's [collective-use computer centers], where the volume of data processed is estimated to be in the range of 10^{12} bytes. For example, [5] proposes the development of fundamentally new external memory types and/or new data organization techniques for creation and efficient operation of the data base of the VTsKP as a way out of the situation.

This work proposes a new method of organizing data in the external memory of the computer, which will enable creation of large files and data bases with a capacity that, if not infinite, is at least several orders greater than that of multiple-volume data sets. This method continues the ideas previously used by the author for simulating the virtual external memory in the OMEGA operating system, realized on the M-222 computer [6].

Formulation of the Problem. Comparing existing data organization methods [3], it must be noted that the memory with direct organization is used most effectively (lowest loss for service data). This loss rises with a library organization, but the most convenient access method is provided: by section name. The memory is used most wastefully with an index-serial and serial organization. We should note that only the serial organization is allowable for any type of external memory unit.

Considering the simplicity of the library method of access (by names), the efficiency of the direct method and the universality of the serial one, the conclusion can be drawn that a solution to the problem of realizing the virtual external memory is possible by creating a new data organization method, combining the advantages of these methdds yet without their shortcomings.

This new data organization method must satisfy the following conditions: it must be uniform for different external memory devices, and provide access to any memory type; finding data on the magnetic disk or magnetic tape storage must be done by name, which will allow virtual access; the direct access memory must be used with maximum efficiency (i.e., minimum system loss). Since the information on the data should be stored on a magnetic disk storage, its inefficient use considerably reduces the total system capacity; all the external memory must be distributed dynamically [7], although the consequences of "recording with a jump" must be considered for the magnetic tape. Otherwise, the memory fields must be periodically compressed (for example, as a result of purging unneeded files), which sharply

reduces overall system efficiency; and the data organization method must not allow the user to set aside an empty external memory for future data, as is done with individual distribution; i.e., it must be similar to the library system, where the memory is not reserved beforehand.

Data Organization Method. The requirements above are not met by any existing methods. In developing the data organization method, the foundation used was the library one, which is almost unused, except for operating systems [8].

With the new method, all the external memory can be represented as a semi-continuous tape, divided into sections of identical length equal to the length of the magnetic disk track (3625 or 7294 bytes). The beginning of the tape (Fig. 1) are the capacities of the magnetic disk used; capacities of the magnetic tape reels are attached later.

The first six magnetic disk sections contain the lists of names and the first section of the catalog containing data on the occupied and free external memory segments. Then, beginning with the first byte of the next section, the memory is dynamically distributed for the user data sets by allocating memory segments when the data are loaded. The length of each segment is determined by the number of bytes of the data loaded.

The memory tape is thus a dynamically segmented region (DS-region), with the empty DS-region consisting of one segment. The DS-region is one of collective use, similar to the library file. But the library method of data organization is supported by the serial method of access, whereas the DS method is based on the direct access method REGIONAL (1) for disks (which completely eliminates loss for system information if the section length equals that of the disk track); the serial method of access is used for magnetic tapes, preserving the same section length for commonality of processing.

The user data sets are thus written inside one file REGIONAL (1) or one file CONSECUTIVE (for tapes). The file of the DS-region is created beforehand on disk, similar to the manner in which a library file must be created beforehand. The catalog of the DS-region performs the role of library directory, but extended to sets on magnetic tape. In this connection, dynamic distribution of the free DS memory is provided for the new catalog sections, since the capacity of one section can be clearly insufficient for such a large region of a multilevel memory.

An important feature of the DS method of data organization is the absence of gaps between the end of one data set and the beginning of the next one, since the catalog parameters define the relative number of the section of the DS-region where the set starts, the number of the byte of the beginning of the set inside this section, and the total data set length.

Reading and writing of the data set are done by exchanging sections between the buffer of the working memory access programs and the external memory region: during loading into the working memory, the file is "assembled" from fragments going to various sections but arranged serially in the external memory, which provides a high exchange rate; during output to the external memory, the file is broken down into fragments, which are placed in adjacent external memory sections. Before the beginning of output from the disk, the last, incompletely filled section is loaded,

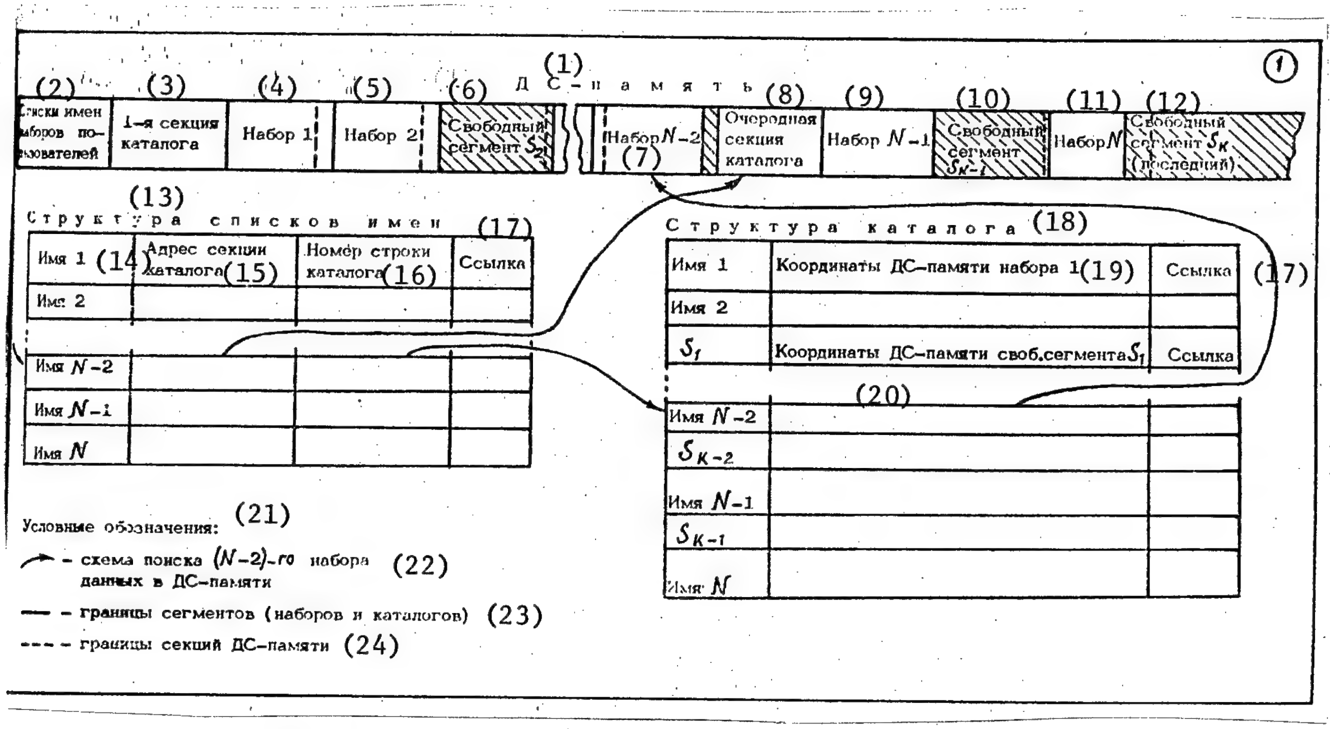


Fig. 1. External Memory Tape

Key:

1. DS-memory
2. Lists of names of user sets
3. First catalog section
4. Set 1
5. Set 2
6. Free segment S_2
7. Set N-2
8. Next catalog section
9. Set N-1
10. Free segment S_{k-1}
11. Set N
12. Free segment S_k (last)
13. Structure of the lists of names
14. Name [1,2,...]
15. Address of catalog section
16. Number of catalog string
17. Reference
18. Structure of the catalog
19. Coordinates of DS memory of set 1
20. Coordinates of DS memory of free segment S_1
21. Key
22. Search scheme of (N-2) data set in DS memory
23. Boundaries of segments (of sets and catalogs)
24. Boundaries of DS memory sections

and the first file fragment added to it (until the section is filled, or until the end of the file). The section thus filled is returned to the appropriate external memory volume, and the process is repeated until the end of file output.

The DS method of data organization thus has fundamental differences relative to the library method, despite the outward similarity.

First. In the library method, the memory of the disk region is divided into fixed blocks, and each array occupies a finite number of them. An incompletely occupied block cannot be used for another array, which lowers the filling factor. The DS method uses the entire memory for user data sets.

Second. The library method is only intended for magnetic disk storages, whereas the

DS method is hardware-independent. In fact, any types of external memory devices (including those not yet developed) can be represented as a series of sections of fixed length: for magnetic drums, they are sections; for magnetic disks, tracks; for tapes, records or regions (for second-generation computers). The advantage of the DS data organization method lies in the fact that it is theoretically applicable for processing data on different external memory types, which enables its use for utilizing the entire field of the external memory (more accurately, the given set of volumes of various types of external memories) as a very-large-volume virtual external memory.

Third. In the library data set, the memory freed after purging unnecessary sets cannot be reused until the entire library is compressed. The DS method provides the possibility for immediately reserving the free region, since it goes to the list of free segments processed by the catalog management program. Selection of the location for the next data set is done by searching in the list, resulting in dynamic distribution of the external memory. Reorganization of the memory with such a distribution method is not needed until at least 95% of the DS-region's capacity is filled [9].

Fourth. The library file cannot occupy more than one direct access volume; the DS-region can occupy the entire external memory area regardless of the number and type of media.

Fifth. Only similar data sets may be written into the library file, namely: initial, object or load modules. Any data may be put into the DS-region file: packets of jobs (of the catalogued procedure type), initial modules, data arrays used in problem programs, intermediate and result data, test examples, documentation, etc.

Sixth. Reference to the library file is only possible through the control operators of the job control languages, or from programs written on the assembler. Reference to the DS-region is possible from a problem program written in PL/1. The problem program can obtain access to any byte of the dynamic region of the DS memory (not just to a certain data set), which in some cases enhances the exchange efficiency and creates the possibility for expanding the functions of the DS-region control system by developing new control programs of access methods.

Realization of the Data Base Using the DS Method of Memory Organization. Based on the method considered of organizing data on magnetic disks and tapes, a data set control system (SUND) has been developed: PROTON. This SUND version is realized in the YeS OS medium, and designed for creation, on-line management and updating of a data base consisting of two-dimensional data sets.

The PROTON SUND is multifunctional, and can be used as: an automated fund of algorithms and programs on magnetic tape or disk; system of automated synthesis of texts of programs in the fund; file of numerical data, texts, and working user sets; and system of processing structured data, consisting of records of fixed length for automated production control systems.

The necessity and importance of creating such systems is emphasized in [10,11]. In fact, the PROTON SUND is an intermediate step between utility programs of operating systems and data banks, providing control at a rather high level of jobs and data of user groups.

Realization of the DS method on disk provides memory use of over 99.5% (for each data set, a string in the catalog and element in the list of names are required--42 bytes in all). With standard methods of data organization, the disk is not used more than 90%. Text data (initial modules, jobs, etc.) are compressed when loaded into the base, which increases the recording density by a factor of 2.4 (this coefficient's value is taken from the results of writing 426 sets of text data in 1980-81). Consequently, for text data one pack of magnetic disk storage, marked out for the DS-region, replaces three packs with the standard processing mode. A basic contradiction is also eliminated: individual distribution of memory and the collective nature of external memory volume use. In other words, memory does not need to be reserved, since the entire free part of the DS-region is accessible.

Since the memory is only distributed for the actual volume of data, the SUND has a large set of editing, correction and updating resources, substantially exceeding the capabilities of existing YeS OS utility programs. The main problem here is the increase in the size of data sets. This is solved by automatic division of the set into segments not exceeding 32K byte in length, which are connected into a list by pointers. This solution removes limitations on data set length, and enables processing of only one segment during editing (updating), by reducing the access time.

Questions of protecting data from unsanctioned access are solved in the PROTON SUND: data may be updated only from lists of set names automatically compiled when loading the data sets. Each user or group identified by the system by the access password thus has a list of names for updating. Data sets not on this list may only be read. System data sets (catalog, lists of names, password file, etc.) are only accessible for reading by a special dynamic key.

The SUND software perform statistics of references to user data sets: total number of references, average period and data of last reference. If the data base regions on magnetic disk are full, the redistribution facilities rewrite onto the magnetic tape the files having the greatest access period. Freed space on the disk is reserved for new data. Upon access to a file on the tape, it is rewritten to the disk (activation), replacing the oldest file on the disk. The PROTON SUND thus has resources for self-organization, which in the operating process gradually divide the region of DS memory into on-line (magnetic disk) and archive (magnetic tape) regions.

When working with a tape, the SUND does not use dynamic distribution of free DS-region segments, since during writing onto the tape the next segment occupied by data can be destroyed (given a malfunction). The segments on magnetic tape freed by set activation are therefore not reused, and are purged from the catalog. This ensures that the next archive set is written at the end of the tape, which the system views as the only existing free segment of the DS-region. Dense packing with the preceding adjacent set is preserved, since the incompletely occupied section must be read into the working memory beforehand; if a tape malfunction results in a record with the malfunction section omitted, then the end of the preceding set's data will not be lost. The SUND distinguishes the disk and tape volumes by the catalog of data sets, in which the code of the volume is written. Empty spots on the magnetic tape are eliminated by the service routine for duplicating the magnetic tape volume, with simultaneous correction of the SUND catalog.

Structure and Composition of the SUND PROTON. Five data processing levels can be singled out in this SUND (from the standpoint of the problem and system user): service; problem; structural; level of data organized by the DS method; and level of physical data segments (Fig. 2).

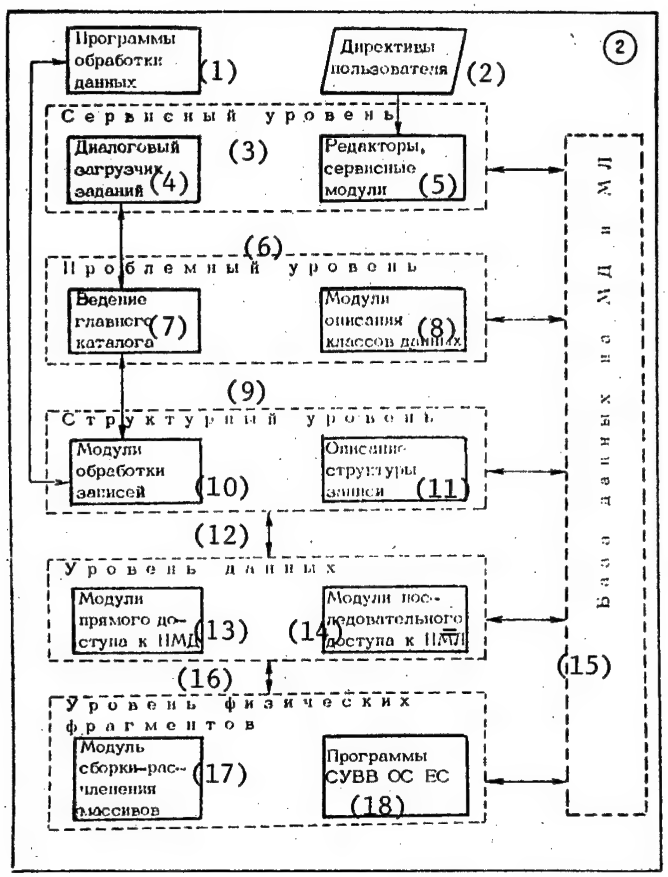


Fig. 2. Data Processing Levels in the SUND

Key:

1. Data processing program
2. User directives
3. Service level
4. Dialog job loader
5. Editors, service modules
6. Problem level
7. Main catalog management
8. Data class description modules
9. Structural level
10. Record processing modules
11. Record structure description
12. Data level
13. Modules of direct access to mag. disk storage
14. Modules of serial access to mag. tape storage
15. Magnetic tape and magnetic disk data base
16. Physical fragment level
17. Array assembly-disassembly module
18. Programs, SUVV YeS OS

The service level works with data sets located in the DS-region as jobs for the operating system to execute the corresponding tasks or process data (like the catalogued procedures in the YeS OS). There is a simple form for describing such data in the data base. The service level is supported by user interface with the terminal through the monitor program in the dialog mode. The input for the given level is the name of the task and the values of the variable parameters of the YeS OS job control language; the output, formulation of the jobs formed into a queue for execution.

The problem level reflects a data difference from the problem user standpoint. It is provided by management of the main catalog, and special lists and tables. User access to this level is possible from problem programs of access to modules GAMRDR and GAMWTR, with indication of the data class, set name, exchange mode, segment length, and displacement relative to the beginning.

The structural level, supporting the logical structure of the data, provides partial independence of data description of the problem programs, and is written by structures in the form used in PL/1. User connection with the structural level is provided from problem programs of access to the module LEVEL, enabling logical records to be processed by specified values of the keys, in series or in the ascending order of the keys, as well as sorting of logic records.

The level of data organized by the DS method in the multilevel external computer memory enables the logical structure of the data to be separated from their physical organization. It is supported by the direct access interface on the YeS OS magnetic disk, and serial access with preliminary supply of the record on magnetic tape.

The level of physical segments provides effective use of external memory (100% of the magnetic disk capacity, minus 42 bytes for table fragmentation for each set), and independence of the physical size of the sets from the size of the DS memory segments. It is supported by the PROTON SUND interface and the data array assembly-disassembly module.

All the system modules are written in the assembler and PL/1. The volume of the system's program is 460K bytes (93,000 instructions). The labor input of the generation is 30 man-years; taking into consideration the dialog operating mode [12], 8 man-years.

Management of the Data Base Catalog and Search. When data are loaded into the base, the file is automatically catalogued in the SUND catalog. The catalog's structure is such that it contains not only the information required for identifying and fetching files (name, length, beginning address, data class, record length), but also a two-way list of free segments appearing when files are purged or automatically transferred after editing (as a result of the increase in length), together with a list of segments of DS memory in the ascending order of addresses and memory levels (list of connectedness).

The same catalog string can belong to a list of free or occupied segments, depending on the value of the references (pointers) situated in certain positions. A second pair of references includes this string in the list of connectedness of the DS memory. This catalog structure completely eliminates the need for its reorganization.

The catalog consists of sections, which the SUND views as segments of the system set of data. The first section has a fixed place in the DS memory and is formed at system generation. After it is filled, the second section is automatically formed on the free disk section, etc. The catalog capacity is thus limited only by the number of sections that can be situated on the disk pack. For a 29M byte magnetic disk storage, the catalog can contain up to 600,000 strings (names), which essentially covers the practical needs of any data processing systems.

The "separate chain" method described in [9] is used to eliminate multiple access to the disk to find a file name in a multi-section catalog. In parallel with the catalog management, the system forms chains of lists of the names of data sets with pointers to the key of the catalog section containing complete information on the set, and to the number of the string in the section. Therefore, instead of a search in the catalog there takes place a search in separate lists, whose average length is:

$$L = CEIL(N/1000),$$

where N is the total number of data sets in the entire DS-region.

The name lists are also stored in sections, and generated dynamically by the system. To speed up the search for frequently used segments of data sets, the method of controlled input to the list is used, in which the name of a set with frequent access is placed at the beginning of the list [9].

Consequently, for any data set only two disk reading operations are required (list+catalog) given any value of N, which increases the speed of PROTON SUND.

Evaluation of Data Base Efficiency. Realizing the data base by the memory organization method proposed enables several problems to be resolved at once.

The capacity of the data base is increased to $1.8 \cdot 10^{10}$ bytes, given a DS memory structure with one 29M byte disk. The capacity doubles with each additional magnetic disk storage added.

Complete automation of memory distribution on magnetic disk and tape is provided, together with the possibility for access to the DS-region from problem programs. This frees the programmer from work related to file description, distribution of peripheral devices, selection of the data organization methods and access to them, error correction in estimating the memory needed, and so forth.

The dialog mode of communication with the SUND considerably improves user labor productivity, even without displays (dialog from the operator console). An operating mode has been realized with the YeS7927 display, controlled by the version 6.1 YeS OS.

The variety of data and their different description, along with an identical physical organization and uniform access method, ensures the multifunctional nature of the PROTON system, and its convenience and ease of use.

A low access time is ensured (1-10 seconds for magnetic disk), which depends only on the segment size, not on the status of the data base.

The simple operation and self-organization facilities of the base enable it to be introduced in several days.

The PROTON SUND is realized on the YeS1022 with a minimal configuration: 256K working memory; one 7.25 or 29M byte magnetic disk storage; one magnetic tape storage. The system is controlled by the YeS OS (version 4.1 and above) in MFT and MVT modes. The magnetic tape storage is not required until the region reserved for the data base on the disk is full. A dialog mode requires 120K of memory; access from problem programs, 30K for access modules and 14-20K for buffers.

BIBLIOGRAPHY

1. Pashchenko, S.N., Pelipenko, N.I., "Program Control of Memories on Magnetic Tapes", USiM, 1980, No 2, pp 73-76.

2. Zhimerin, D.G., Myasnikov, V.A., "Avtomatizirovannyye i avtomaticheskiye sistemy upravleniya" [Automated and Automatic Control Systems], Moscow, Energiya, 1975, 680 pages.

3. Devis, U., "Operatsionnyye sistemy" [Operating Systems], Moscow, Mir, 1980, 436 pages.

4. Konovalov, N.A., Kryukov, V.A., Lyubimskiy, E.Z., "Controlled Virtual Memory", PROGRAMMIROVANIYE, 1977, No 1, pp 28-36.

5. Maksimenko, Yu.V., "Evaluation of Technical and Economic Characteristics of Automated Data Banks", in "Vychislitel'nyye tsentry kollektivnogo pol'zovaniya" [Collective Use Computer Centers], Moscow, Statistika, 1979, pp 240-256.

6. Semchenkov, L.I., Kokhanov, V.M., "Data Control Facilities and their Realization in the OMEGA Operating System for the M-222 Computer", in "Razrabotka i vnedreniye ASU v narodnom khozyaystve Kirgizskoy SSR: Materialy Respubl. nauchno-tekhn. konf." [Development and Introduction of Automated Control Systems in the Economy of the Kirgiz SSR: Materials of the Republic Scientific and Technical Conference], Frunze, Ilim, 1974, pp 49-51.

7. Yurchenko, A.S., "An Algorithm for Dynamic Segmental Distribution of Memory", KIBERNETIKA, 1981, No 1, pp 45-54.

8. Chukin, Yu.V., "On Selection of File Organization", USiM, 1977, No 3, pp 74-78.

9. Knut, D., "Iskusstvo programmirovaniya dlya EVM. Osnovnyye algoritmy" [The Art of Computer Programming. Basic Algorithms], Moscow, Mir, 1976, Vol 1, 736 pages.

10. Lipayev, V.V., "The Status and Problems of Production of Software for Control and Data Processing Systems", USiM, 1980, No 1, pp 9-15.

11. Shkurba, V.V., Mandrusova, G.P., Manovich, S.F., Opanovich, M.I., "The AL'FA-Interface of User With Computer and Resource for Integrating ASU Information Systems", KIBERNETIKA, 1980, No 6, pp 8-13.

12. Lipayev, V.V., Minayev, M.A., Chichigin, A.B., "Characteristics of Dialog Interaction of Programmers with Computers in Generating Control Programs", USiM, 1979, No 3, pp 34-40.

9875
CSO: 1863/140

UDC 681.3.06

PACKAGE FOR CREATING HARDWARE-INDEPENDENT SOFTWARE FOR MACHINE GRAPHICS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83
(manuscript received 4 Mar 82, after revision 13 May 82) pp 115-117

[Excerpts from article by G.A. Pankeyev, "A Package for Creating Hardware-Independent Software for Machine Graphics"]

[Excerpts] As facilities for graphic input and output of information developed, designers had to solve the problem of creating software that is independent of the particular features of given devices. This question was resolved fairly long ago and successfully [1,2], but the introduction of dialog resources for work with display units has led to an attempt to more fully utilize the technical and functional capabilities of the display units [3,4].

The existing variety in input hardware has been standardized by defining a certain set of virtual input devices. Graphic protocols subsequently developed [5,6] have enabled a certain standardization of graphic information transmitted through the network, and its representation in a hardware- and machine-independent form. One goal in developing software for dialog operation with graphic devices is the possibility of creating hardware-independent application programs [7-10].

One of the first Soviet developments in this direction was the package of graphic subroutines ANEGRAF, created at the Novosibirsk branch of ITM [expansion unknown] and the Computer College of the USSR Academy of Sciences. Its creation utilized ideas of graphic protocol and virtual input devices.

The machine independence of the package is achieved by using FORTRAN and the fact that information is exchanged between the hardware-independent part of the package and the terminal program in the graphic protocol format, which is machine-independent.

The package currently provides the application program with the capability of working with different displays, connected either directly to the main machine (YeS7064 to BESM-6) or to the terminal computer (SIGD to M6000). In the former case, the terminal program is in the main machine; in the latter, in the terminal one. The possibility is provided for working with two displays and the YeS7052 plotter simultaneously.

Conclusion. Operation of the ANEGRAF package and creation of a dialog graphic system based on it have shown that the package's resources enable creation of hardware-independent application programs.

In the realization on the YeS7064, all the virtual input devices described could be represented for concrete physical devices fairly conveniently and simply.

The package's shortcomings would include its large volume, related to the presence of operations of work with elements, and the need for the terminal program to store the display file, to provide the possibility of displaying the picture regardless of the moment of its creation.

In conclusion, it can be noted that the organization of the ANEGRAF package as hardware-dependent and independent parts with exchange of graphic information between them in the graphic protocol format is especially convenient, and in some cases necessary in organizing computer and collective use computer center networks.

9875
CSO: 1863/140

UDC 681.3.06

GRAPHIC PACKAGE FOR DATA ANALYSIS: STRUCTURE AND MAIN PRINCIPLES

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83
(manuscript received 14 Sep 81, after revision 7 Jan 82) pp 111-114

[Excerpts from article by L.G. Kaminskiy, S.V. Klimenko, V.N. Kochin, A.V. Samarin and A.P. Sokolov, "A Graphic Package for Data Analysis. Structure and Main Principles"]

[Excerpts] Introduction. Our previous works [1-3] have examined the features of using machine graphics in experimental physics, and noted the main specifications of machine graphics software designed for tasks of data analysis and on-line monitoring of the work of experimental facilities. The range of use of machine graphics for representing final results (of an experiment, computations, etc.) in graphic form has been labelled illustrative graphics [4], which stresses its difference from interactive use of machine graphics, allowing not only graphic output, but also graphic input. However, this does not mean that illustrative graphics are not interactive. The loading resources are also used, but mainly for controlling the output, not for loading images.

Research performed and the experience of using machine graphics in the IFVE [Institute of High Energy Physics] have shown that the most appropriate form of machine graphics software for such tasks is a special-purpose graphic package of subroutines. The package must provide terminal independence of the application programs, and meet requirements of mobility and expandability of the functional capabilities.

This paper describes the basic principles, structure and features of realizing the problem-oriented package ATOM, designed for use in problems of analyzing experimental physics data.

Development of the package used the recommendations of ACM/SIGGRAPH (Core Graphics System) [5,6], reworked in consideration of the special features of machine graphics use in data analysis problems [3,4,7,8].

Conclusion. The ATOM package described is used in the IFVE on the ICL-1906 and DEC-10 computers. It enables operation with the TEKTRONIX-611, 4010, 4012 and 4014 displays, ICT-1934 (CALCOMP), SERVOGOR-211 and WX4636 (WATANABE) plotters, and the use of alpha-numeric displays and printers as simulators of graphic devices. Fast mastery of the package by an expanding number of users and its efficient use for various tasks solved at the institute indicate the correctness of the concepts used.

9875
CSO: 1863/140

UDC Δ65.011.56:621.951.1

PROCESS CONTROL SYSTEM FOR DRILLING PRINTED CIRCUIT BOARDS

Moscow MEKHANIZATSIYA I AVTOMATIZATSIYA PROIZVODSTVA in Russian No 2, Feb 83
pp 27-28

[Article by engineers S. D. Perekhvatov, A. F. Abramov, V. P. Shlegel' and I. D. Orlov]

[Text] Computers are now widely used in controlling manufacturing processes. An example is the printed circuit board [PCB] drilling control system described below.

The system controls a section of 15 drills with numeric control [NC]. The types of tools are: domestic, S-128M; foreign, Alpha-Z (United States), Multifor-II (Switzerland) and ABL-24 (FRG).

Modes of tool control are: for the domestic, direct control from computer (through a non-standard unit for machine tool control); for the foreign, combined control from a computer (control by computer by using a machine tool control block (BUS) and an NC unit).

The system allows: simultaneous control of positioning and drilling; sequential drilling of batches of boards of the same type placed on the tool table (maximum number of batches is 32); rescaling of coordinates of the control program as a function of the value of the adjustment input by the machine operator with angular displacement of the pattern relative to base holes; output of a control program on perforated tape; monitoring the operation of each tool control channel by using the central dispatch console (PTsD); output to alphanumeric screen display of on-line processing information for one of the 15 tools when the system operator calls; output of accounting information at the end of the shift on an alphanumeric printer; and preparation of accounting information for higher levels of management.

Figure 1 is a diagram of the PCB drill control system. Shown schematically in the diagram is one machine tool control channel for each type: I is the S-128M, II is the Alpha-Z and ABL-24 and III is the Multifor-II. The process control system has one level. A specialized complex based on the SM-1 No. 4 control computer complex is used as the control computer complex (UVK).

The intermediate device between the control computer complex and the equipment in all tool control channels is the BUS [machine tool controlblock]. The control computer complex is connected to the machine tool control unit through a nonstandard interface module (MS). If the control computer complex malfunctions, the tools are controlled by standard NC units.

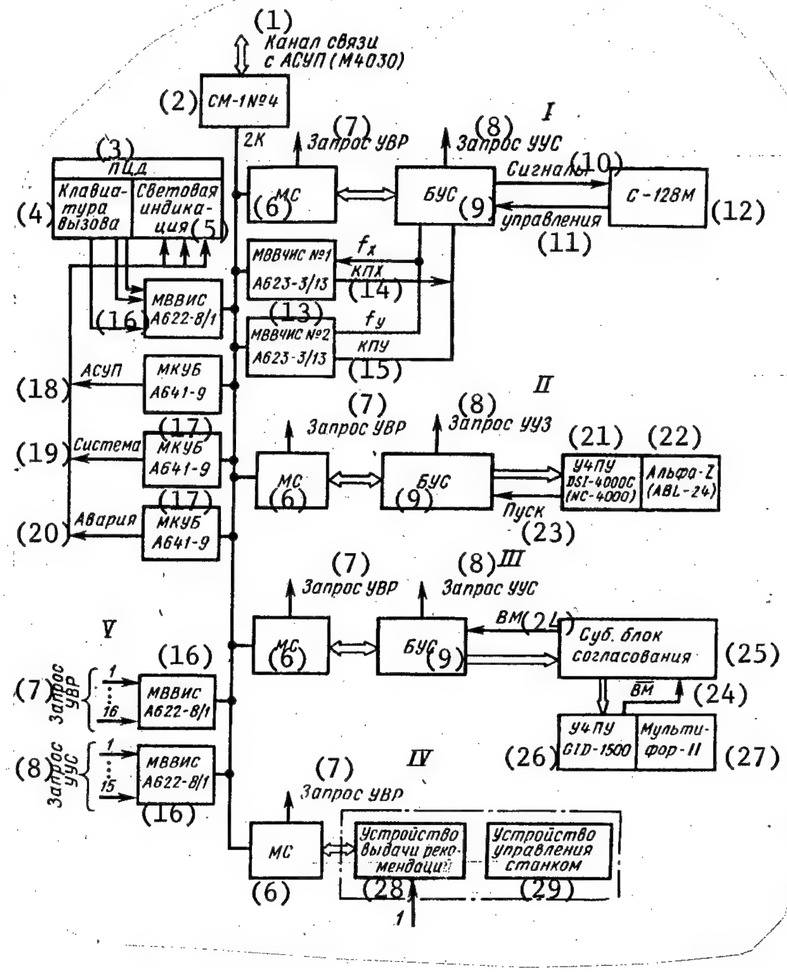


Fig. 1. Diagram of Printed Circuit Board Drilling Control System

Key:

1. channel for communication with ASUP [automated production control system] (M4030)
2. SM-1 No. 4 control computer
3. PTsD [central dispatch console]
4. keyboard for calls
5. light display
6. MS [interface module]
7. inquiry to UVR [device for issuing recommendations]
8. inquiry to UUS [machine tool control unit]
9. BUS [machine tool control block]
10. control signals
11. control signals
12. S-128M
13. A623-3/13 MVVChIS [numeric pulse signal input modules] No. 1, 2
14. KPx [X positioning end]
15. KPy [Y positioning end]
16. A622-8/1 MVVIS [initiative signal input modules]
17. A641-9 MKUB [not further identified]
18. ASUP [automated production control system]
19. system
20. emergency
21. DSI-4000C (NC-4000) UChPU [NC unit]
22. Alpha-Z (ABL-24)
23. start
24. VM [signal for sending entire frame of control program]
25. interface subblock
26. GID-1500 NC unit
27. Multifor-II
28. device for issuing recommendations
29. machine tool control unit [UUS]

General principles of system operation are as follows. The NC tool operator uses the keyboard on the front panel of the BUS [machine tool control block] to key in the decimal number of the PP [printed circuit board] and the drill diameter (or the operator obtains this information from the UVK [control computer complex], and the quantity of boards placed on the tool, and then enters this information into the control computer complex. When the control program is in internal memory for the complex, the operator loads the machine tools and presses the "Start" button to send the control program to the BUS [machine tool control block].

Information on the value of transposition by the coordinate $\pm X$, $\pm Y$ is written to the A623-3/13 MVVChIS [numeric pulse signal input modules]. Pulses go to the input for the modules from single-rotation couplings on the machine tool (fx, fy). Upon overflow of the module counters, the signals "End of X Positioning" (KPx) and "End of Y Positioning" (KPy) are generated; they are processed by the BUS [machine tool control block] according to a specific algorithm, generating the "Drill" command. Information on the next hole ("Inquiry to Machine Tool Control Unit") is requested by the signal "End of Drilling."

The NC unit for the tools generates a "Start" signal, meaning a request for a row of the control program. The BUS [machine tool control block] sends the signal "Inquiry to Machine Tool Control Unit" upon which the control computer complex sends bytes of information from the control program to the DSI-400C (or NC-4000) NC unit.

The NC unit for the Multifor-II machine tool generates a request for an entire frame of the control program (a "VM" signal). In the interface subblock, this signal is converted by level and goes to the BUS [machine tool control block], where upon a "VM" signal, an initiative signal "Inquiry to Machine Tool Control Unit" is generated for each row of a frame of the control program. As soon as an entire frame of the program is sent to the NC unit, the "VM" signal is removed and the frame is then processed by the NC unit itself.

The information 1 on finished products is entered through channel IV also by using a BUS [machine tool control block], at which a device for issuing recommendations and the OTK [department of technical control], installed at the controller's work station, are installed. Information on the decimal number and on the amount of boards that have passed the department of technical control is entered into the control computer complex to prepare accounting information for higher levels of management and for output to the printer forms at the end of the shift.

Channel V for input of initiative signals consists of two standard modules, part of the nomenclature for the SMEVM [System of Small Computers], of the type MVVIS [initiative signal input modules], A622-8/1.

The PTsD [central dispatch console] is controlled by three A641-9 MKUB [not further identified] modules and one MVVIS [initiative signal input module]. The process engineer-operator from the central dispatch console can call to the alphanumeric display screen online-process information for each of the 15 machine tools included in the system and monitor the operation of each channel for machine tool control.

The hardware is located as follows: the control computer complex with external devices and the central dispatch console are installed in a machine room; the modules controlling the central dispatch console are placed in an input/output interface (SVV, A151-6); and the BUS [machine tool control block] is at the machine tool operator's workstation. Interface modules and MVVChIS [numeric pulse signal input modules] are placed in two splitters of multiplex interfaces (RIM, A714-5/1, A714-5/2), which are in the drilling section in the distributing cabinet. Also here are the power supplies and distributing panels with the use of which all necessary connections are made. The distributing cabinet can be up to 3 km away from the control computer complex.

The SM-1 has general and specialized software. Used as general software is the DOS ASPO [modular software system disk operating system]. Special software allows monitoring information coming from operating personnel.

Information support for the system provides for the capability of expanding information files with regard to increasing the number of machine tools.

Programs supporting process control functions are divided into three groups: control of drilling, preparation of control programs, and printing information on the machine tools per shift and output of this information to ML [magnetic tape] for transfer to the process control system.

All three groups of programs are assembled as separate load modules.

This system was developed at the Ryazan Process Design Institute and is now being introduced at the Orel Plant for UVM [process control computers].

System introduction experience will allow defining the possibilities for improving system performance and expanding service functions for preparation and adjustment of control programs.

8545
CSO: 1863/160

UDC 65:338:658.51

IMPROVING SPECIAL-PURPOSE PLANNING FOR DEVELOPMENT OF A MANAGEMENT INFORMATION SYSTEM FOR THE NATIONAL ECONOMY

Moscow MEKHANIZATSIYA I AVTOMATIZATSIYA PROIZVODSTVA in Russian No 2, Feb 83 pp 28-30

[Article by engineer Z. V. Antoshina]

[Text] Expansion of efforts on automating management of the national economy, an increase in outlays for research and development, the introduction and operation of management information systems and the need for further improvement in their efficiency have caused rapid development and application of the methods and facilities of special-purpose program planning where one can trace distinctly three stages.

In the first stage, guidance for design of management information systems was implemented on the basis of the Coordination Plan for Solving Major Scientific and Technical Problems in Development of Management Information Systems. In the 7th and 8th Five-Year Plans, plans were drafted by the USSR State Committee for Science and Technology under the USSR SM [Council of Ministers] on the basis of suggestions from the USSR Academy of Sciences and USSR ministries and departments. The organization and methodology of developing the Coordination Plan basically in that period enabled solving the posed problems of developing management information systems.

At the same time, the Coordination Plan was just the sum total of measures implemented by ministries and departments on design of management information systems [MIS]. It did not achieve the proper coordination and tie-in of stages of efforts, composition of tasks for MIS functional subsystems and their development schedules. What was implemented was just the formalization of the separate tasks of planning, accounting and monitoring which had to operate in the standalone mode "unincorporated" into the process of planning and management. MIS design, as a rule, was performed in isolation from efforts on automating management of territorial organizations, enterprises and associations and without due regard for features of development of sectors in the national economy and the change in their managerial structure.

The second stage in development of methods and organization of special-purpose program planning for MIS development in the 9th and 10th Five-Year Plans was characterized primarily by broad solution of the scientific and technical, industrial, organizational and economic, social and other problems in MIS development and introduction, i.e. by a complex of measures.

The main task of the program in that period was to realize the requirements of organizational, methodological and technical unity of systems designed and developed for subsequent unification of them into the statewide system for information acquisition and processing for planning and management of the national economy.

A feature of this stage in MIS development was special-purpose orientation toward ensuring close coordination of system design and introduction with the entire complex of measures on improving management, as well as coordination with the general schemes for management of the sectors in the national economy. In an ever greater number of MIS's, management tasks began to be selected from the positions of the systems, problem-oriented approach, and the percentage of tasks for optimal planning and forecasting yielding the highest economic effect was increased. Implemented were statement and solution of multilevel optimization problems for management of sectors and the switch to progressive forms of designing systems based on standard solutions.

Efforts were carried out to tie individual tasks together based on unified technology for planning, accounting and monitoring. All this created the capability of incorporating tasks ready for operation directly in the managerial process.

Work on preparing source information was considerably reduced through reducing nonautomated tasks, creating systems for information storage, updating and retrieval, and generating a standard reference base.

One of the main factors which hindered further application of programs on solving scientific and technical problems of MIS development and which reduced their efficiency was the incomplete coverage by it of all the stages in the life cycle of systems. The main portion of efforts on introduction and operation of MIS tasks and facilities was performed outside the scope of the program and not coordinated from a single center.

At that time, it became necessary to ensure within the scope of the program a purposeful, coordinated and complex design of MIS's (there were substantial differences in the level of design of individual subsystems, complexes of tasks and MIS facilities enabling their functioning).

Uncoordinated development of MIS's did not yield in full measure high economic, scientific and technical and social results from the operation of them.

The latest stage in further improvement of the entire system of planning and management, including MIS development, was established at the 26th CPSU Congress where this task was assigned: "Ensure further progress and enhance the efficiency of the MIS and multiuser computer center network while joining these systems into the unified statewide system for information acquisition and processing for accounting of planning and management."[1]

To carry out this task, it became necessary to coordinate MIS's at all levels and primarily the ASPR [automated system of plan calculations], the OASU [industrial

---

[1] "Materialy XXVI s"yezda KPSS" [Materials of the 26th CPSU Congress], Moscow, Politizdat, 1981, 123 pages.

sector MIS's] for the ministries and departments, MIS's for territorial administrative agencies, ASUP [enterprise MIS's] and other systems used in drafting long-term and current state plans for USSR economic and social development. Also, autonomy in solving intersectorial problems of planning, accounting and monitoring had to be eliminated, and comprehensive coordination of them had to be ensured on the basis of compatible enabling facilities with extensive use of data transfer systems and standard documentation systems.

Now playing a major role in solving the problems posed is the complex program for MIS development where development of methodology for drafting it holds a prominent place. The main difference between the complex program and that for efforts on solving scientific and technical problems is the intensification of comprehensiveness of planning based on including and coordinating in a single entity the tasks covering all stages of a system's life cycle. In final form, this program will be a document in which goals are formulated, tasks and types of efforts are defined, coordination of the purpose and resource parts of the program is implemented, the composition of executors and measure implementation schedules are established, and the economic effect and major technical and economic parameters of results for program tasks are defined by stages of MIS development. The specific properties and requirements of the object of planning as well as the scientific and technical and the socioeconomic conditions for developing it must also be considered. These include the following:

--alternatives for plan solutions shaped by the set of strategies for development of the individual MIS as well as by the interaction of these systems within the scope of the OGAS [statewide automated system] in solving the posed problems of planning, accounting and monitoring;

--alternatives for parameters and criteria of MIS developments as well as alternatives in the plan of efforts on coordinated development of these systems;

--multidimensionality of the effect of MIS development alternatives and the problem of automating national economic management as a whole on the development of sectors in the national economy of the union republics, economic regions and society as a whole;

--the lack of a priori information on basic parameters describing the capabilities of various MIS's and resources needed to develop them. This is governed to a considerable extent by uncertainty of results and outlays for both the efforts to be performed (scientific research, development efforts, system introduction and operation) and obtaining the information needed to draft the plan; and

--the complex nature of the problems to be solved in MIS development which in turn requires coordinating the efforts of various ministries and departments and full coverage of all stages in the life cycle of the development of the system, its elements, major supporting elements and other facilities.

A feature of this stage of efforts on improving organization and technique in special-purpose program planning is extensive application of economic-mathematical methods and models to describe the process of development of the comprehensive program for MIS development, automation of the process of drafting it and monitoring implementation of it.

But use of modeling in special-purpose program planning is not always successful; this is due to a number of methodological, organizational, technical and social causes including, for example, the following:

--the impossibility of developing and above all making efficient use of one general-purpose model, with the use of which one could identify and analyze the various aspects of the MIS development problem;
--the inadequacy of the model describing the special-purpose comprehensive program, and its structural and parametric uncertainty; insufficiently complete consideration of the creative nature of the processes of modeling, its uniqueness and the emergence of a substantial share of new elements in the course of system development;
--the lack of consolidated and aggregated methods for objective assessment of MIS parameters by program developers and their lack of professional and psychological training in applying methods and models of management by using computers; and
--the imperfection in informational, organizational-legal, mathematical and other supporting elements.

These deficiencies can be eliminated to a considerable extent with development and extensive application of facilities for special-purpose informational representation of the problems and processes in drafting the program, developing man-machine procedures for assessment and decision-making, and developing a multilevel system of management models. A major factor in solving the problems posed is the development of the organizational-economic mechanism, introduction of the various estimates for alternatives, suggested by executors, of conditions for comprehensive support of efforts, and development of an incentive system for carrying them out.

One can note the following principles for generating a special-purpose comprehensive program for MIS development:
--the program is generated as a system of interrelated procedures including organizational support (distribution of procedures among executors, procedures for executor coordination), informational support (methods of acquisition, analysis and generalized representation of information) and software (methods of analysis and decision-making);
--program alternatives are generated, and decisions are made in the interaction between the people making the decisions and the experts for establishing the relations and commensurability of solutions to the problem of automating national economic planning and management as well as analysis and consideration of the consequences of these solutions.

The program must embrace a long period which considers all stages in the MIS life cycle, and primarily the concluding stage, i.e. operation of the system, to determine the ultimate effect of its functioning;
parameters being defined in the concluding stage of the plan period (scales of introduction in the stage of operation) are determined approximately, but to an extent adequate for making sound decisions for the immediate plan period and problem as a whole.

Guided by the principles formulated above, one can suggest the following sequence for drafting a comprehensive program for MIS development.

In the first stgae, the organization ordering the system and the lead developing organization jointly formulate the program development specifications including basic indicators, assessment of the main resulting parameters of the MIS to be developed, its efficiency, makeup of problems (tasks) of the program (systems to be developed) and suggested designers.

In the second stage, the designers responsible for individual subsystems (complexes of tasks) prepare proposals on alternative solutions for implementing them, organization of efforts and estimate of resources needed to implement the proposals.

In the third stage, the lead developing organization analyzes and generalizes the proposals of the responsible designers, uses them as the basis for generating program alternatives as a whole, coordinated by resources, schedules for completion and results assumed, and then selects the optimum version.

A program alternative can be selected by proceeding from the values of generalized parameters obtained from conditions of optimality of a comprehensive criterion or a set of criteria.

Optimization of values of comprehensive assessment of the process of MIS development at the stage of operation, including assessment of optimum amounts of introduction of various systems of blocks with regard to interaction of them, can also be implemented by using a comprehensive criterion.

8545
CSO: 1863/160

UDC 65.011.56:629.12

AUTOMATION OF MOLD LOFT-TECHNOLOGICAL PREPARATION FOR HULL PRODUCTION

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA (NAUCHNO-PROIZVODSTVENNYY SBORNIK) in Russian No 4, Oct-Dec 82 (manuscript received 2 Jul 81) pp 11-13

[Article by A.D. Kolchak, engineer]

[Text] One effective way of automating mold loft-technological preparations for the production of hulls for ships is the extensive use of computer technology, mathematical methods and equipment with digital program control (ChPU).

A significant amount of experience has now been amassed in the field of automating laborious mold room calculations, carrying out the technological preparations for hull production, and developing control programs for machinery and machine tools with ChPU. Work on the preparation of control programs was begun simultaneously with the introduction of the first "Avangard" and "Almaz" experimental prototypes of thermal-cutting machines with ChPU.

As the number of thermal-cutting machines available increased--particularly the most progressive ones (of the "Kristall" type)--there was an abrupt increase in the need for control programs. A specialized subdivision, the multiple-user computer center (KVTs), was set up in order to solve this problem.

During the 10th Five-Year Plan, a large number of control programs for thermal-cutting machines and programs for metalworking machine tools with ChPU were developed and released for introduction, the productivity of programmers' labor increased by a factor of 2.8, and the cost of a single program of average complexity was reduced by a factor of more than 2. The economic effect from the introduction of control programs was more than 1.9 million rubles.

These results were achieved primarily because of constant improvement in the production process, the mastery of new programming methods, the modernization of the technical equipment for monitoring control programs, the development and introduction of new forms of planning, the moral and material stimulation of engineering and technical workers, and the development and introduction of a system for quality control at each stage of the production process.

A great deal of importance is attached to systems analysis of the technical and economic data accumulated in the KVTs in connection with the mass output of control programs. It is used as a basis for the determination of norms for all types of expenditures at every stage of the production process and the development of technical

and economic measures for increasing labor productivity, improving quality and reducing the cost of control programs.

The basic document for the development of control programs is the production process chart, the shape and the dimensions of the parts of which are determined manually in the mold loft. The accuracy of the determination of the dimensions of parts, which are also given in the working plan, depends on a multitude of objective and subjective factors. Experience has shown that the dimensions of parts determined by graphic methods differ from the true ones by 2 mm and more; as a rule, the error is on the side of enlargement.

Therefore, the problem of the transition to integrated automation of mold loft-technological preparations for hull production on the basis of a mathematical model of a ship's hull is an extremely urgent one. Such a complex presupposes the solution of the following problems: analytical matching of the outlines of the ship's hull; the laying out of seams and structural lines; the analysis of shell plates; formulation a mathematical model of the ship's hull; the analytical determination of the shapes and sizes of hull parts; the laying out of sheet and shaped metal and compilation of production process charts; calculation of control programs for the thermal cutting of parts; the development of outline sketches for group and unit assembly; the output of information for adjusting beds for the assembly of sections.

The first such complex of projects was realized during the construction of the advanced ore and oil carrier "Boris Butoma." The introduction of the complex made it possible to shorten the cycle of mold loft preparation by a factor of 1.5-2 and the labor-intensiveness of the welding assembly work by 7-10 percent. There was also a significant shortening of the ship construction cycle as a whole and an economic effect on the order of 180,000 rubles was obtained.

Autonomy, which makes it possible automate a certain stage of the work independently of the method used to realize the preceding work, is a characteristic of most of the programs in the complex.

The combination of programs into a single system is made more difficult by the fact that they were developed at different times, using different algorithmic languages for different types of computers. This makes the complex labor-intensive and requires the participation of highly qualified specialists, which causes substantial delays in its introduction on broader scales.

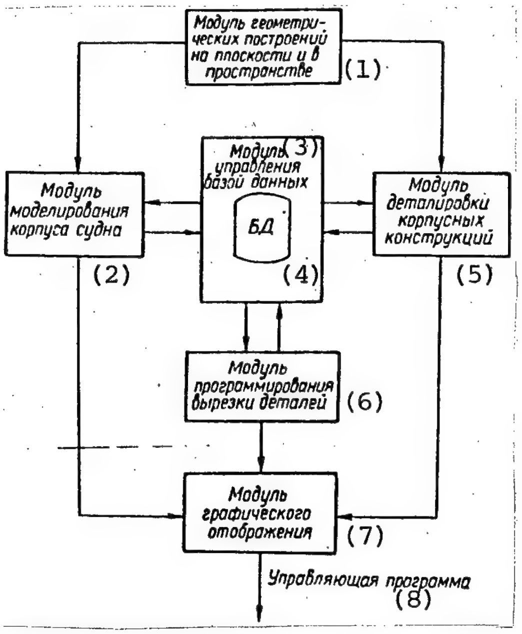
In order to eliminate these flaws in the complex and in connection with the transition to third-generation computers, a system for the integrated automation of mold loft-technological calculations (PLATYeR/DOS YeS) has been developed.

The PLATYeR/DOS YeS system takes care of the modeling of the ship's hull, the detailing of the design of hull structures, and the programming of the cutting of parts by thermal-cutting machines with ChPU.

The figure on the next page is a block diagram of the PLATYeR/DOS YeS system.

Two problem-oriented languages--FORKOS and DIRMON--are used in the system.

FORKOS makes it possible to describe the solution algorithm for problems that take into consideration the specific nature of the designs for a specific ship hull.



Block diagram of PLATYeR/DOS YeS system.

Key:

1. Module for geometric designs (two- and three-dimensional)
2. Ship hull modeling module
3. Data base control module
4. Data base
5. Module for detailing design of hull structures
6. Module for programming cutting of parts
7. Graphic depiction module
8. Control program

The following functions are carried out with the help of the DIRMON language: preparation of the data bank; graphic depictions of the surfaces of the ship's hull, hull parts and hull structures; entry into the tabular information bank and output from it of reference information.

The PLATYeR system's data bank is organized on magnetic disks and contains three data bases: HULL, PART and CHART.

What is stored in the HULL data base is a mathematical model of the hull, which consists of mathematical models of the surfaces (outer plating, upper deck, inner bottom and so forth) and is a representation of the ship's hull as a three-dimensional structure. The PART data base contains mathematical models of the structural parts, whereas the CHART data base contains the mathematical models of the layout of the hull's sheet metal parts.

For the integrated performance of work with the PLATYeR/DOS YeS system, it is necessary to have a YeS EVM [Unified System of Electronic Computers] complex with the following minimum characteristics: a main memory with a capacity of 512 Kbyte; 4 magnetic disk storage units with a capacity of 7.25 Mbyte (YeS-5052, YeS-5056) or three storage units with a capacity of 29 Mbyte (YeS-5061); the standard units that are part of any YeS EVM complex (input from punched cards, output onto punched tape and so on).

The system provides for the output of graphic information on YeS-7054, ITYeKAN-2M and TAKT (with ULPI and LKI interpolators) graphic plotting units, as well as any such unit that is controlled by punched tape in the ESSI format; the output of control programs for Soviet-built thermal-cutting machines with ChPU of the "Kristall" type, foreign machines of the "Sikanat" type and others that can receive information in the ESSI format. It also allows both integrated and independent use of program modules.

At the present time, the system is being introduced industrially in two projects. The system's basic advantages are: a systems approach to the solution of the problems in the complex and the possibility of monitoring the process of determining the shapes and sizes of hull parts, which improves the quality of the detail work (the most labor-intensive process in the complex) substantially and reduces the labor-intensiveness of the assembly work.

11746
CSO: 1863/85

UDC 658.012.011.56:658.512.2

SAPR ASU USERS AND THEIR EXTERNAL WORK SCENARIOS

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA (NAUCHNO-PROIZVODSTVENNYY SBORNIK) in Russian No 4, Oct-Dec 82 (manuscript received 1 Feb 82) pp 14-18

[Article by A.V. Pershin, engineer, and V.M. Yurkov, candidate of technical sciences]

[Text] In accordance with the concept of the automated designing of an ASU [automatic control system] that was developed in the Ukrainian SSR Academy of Sciences' Institute of Cybernetics imeni V.M. Glushkov* and is being used in the "Gorsistemotekhnika" [expansion unknown] NPO [Scientific Production Association] in the creation of a system for the automated planning of territorial links of a republican ASU, the SAPR [automated design system] ASU is understood to be a specialized, automated design organization that realizes innovative industrial technology for the automated designing of ASU's.

The characteristic feature of the concept is its clearly expressed orientation on the user. It is manifested both in the orientation of the SAPR's output product--design documentation--on the specific user (customer) and in the orientation of the technology of the automated designing of ASU's itself on specific participants in the design process (the SAPR director, project leaders, designers of different categories, the administrator of the design information bank and auxiliary personnel). Each participant in the design process works in accordance with his own external scenario, which is a record of the user's working technology that is given to him by the system on a terminal at his own automated working position. The term "external scenario" underlines the fact that in it is shown the direction of the scenario toward the user and the external manifestation of his working technology, in contrast to an "internal scenario," which describes what takes place inside the actually functioning system at one point or another in the external scenario.

Below we describe variants of external scenarios for several categories of SAPR users; these variants reflect the concept of their working technology. On the basis of the DYeKOM demonstration program complex, which was developed at the Ukrainian SSR Academy of Sciences' Institute of Cybernetics and which has (among others) the

*Shkurba, V.V., Meytus, V.Yu., and Yukov, V.M., "System for the Automated Designing of ASU's," in "ASU: avtomatizatsiya proyektirovaniya i modelirovaniya" [ASU's: Automated Designing and Modeling], Kiev, Institute of Cybernetics, Ukrainian SSR Academy of Sciences, 1981, pp 3-15.

Example of User's Working Scenarios in SAPR ASU

User's Instructions	Main Field of Frame
Frame 1. 7 Apr 82 8:45 In order to enter code number, place it in upper left corner of screen and press "VV" [enter] key	Please indicate your code number You may obtain a code number from the system's administrator
Frame 2. 7 Apr 82 8:48 Fill in the table. Upon completion, press "§" key	Primary Questions About Object Questions / Answers 1. Type of production 2. Branch 3. Size of enterprise 4. Number of workers 5. Basic technical and economic indicators
Frame 3. 7 Apr 82 9:00 Call up the next page by pressing buttons "S" and "VV"	Thank you for the information about your enterprise On the basis of the data presented by you, it is possible to give an approximate estimate of the effectiveness of an ASU for your enterprise. The basic effect that the introduction of an ASU in your enterprises can yield is about 1.5 million rubles
Frame 4. 7 Apr 82 9:00 Confirm reception of system report by entering "+" Call up the next page by pressing buttons "S" and "VV"	Project: ASU for X enterprise Project leader: Ivanov, I.I. Stage: developmental planning Completion date: 9 Apr 82 System report: Attention! Completion date expires in 2 days
Frame 5. 7 Apr 82 9:05 In order to select one work or another, enter the appropriate number in the system	Plan development at stage of conceptual designing Compile: 1. Working program 2. Network chart 3. Thematic working plans 4. Users' working scenarios
Frame 6. 7 Apr 82 9:10 In order to select mode, enter the appropriate number in the system	Select operating mode: 1. Teaching mode 2. Standard mode 3. Rapid mode

Frame 7. 7 Apr 82 9:15

In order to call up specification, enter symbol "S" in system

Typical network chart for correlation of design concept to object:

```
          →4—      →7—
1 → 2 → 3       6→ →8 → 10 → 11—
          →5—  ↑   →9 → 12 → 13 → 14
```

Frame 8. 7 Apr 82 9:22

Correct and fill in table, using system's editing facilities. Upon completion, enter symbol "§"
In order to go to next frame, enter symbol "S"

SPECIFICATION

Code #	Name of work	Length (days)	Users
0-1	Make decision on specific technique for inspecting and reproducing appropriate forms		

Frame 9. 8 Apr 82 8:30

Confirm reception of system report by entering symbol "+"
In order to select work mode, enter the appropriate number

Stage: monitoring development
Select work mode:
1. Resume of state of affairs
2. Review of project file
3. Confirmation mode
4. Correct network chart

System reports: Attention! Period for execution of work 4-6 "Correlate Part I of concept model" expires in 2 days

Frame 10. 8 Apr 82 8:33

In order to select document, enter its number in system

Presented for Confirmation

No	Code #	Document Title	Execution Period	
			Plan	Actual
1	211	Purpose	10.04	8.04
2	213	Personnel and external users	6.04	8.04

Frame 11. 8 Apr 82 9:00

Enter number of adopted decision in system

Decision adopted:
1. Return for further development
2. Approved

Frame 12. 8 Apr 82 9:02

Confirm reception of system report by enter symbol "+"

System reports:
Configurational monitoring done successfully
Work approved, entered in design information bank with status "Approved"
Work taken from monitoring by dates

Frame 13. 8 Apr 82 14:00

Ask for text for editing in BPI, base 2
In order to request information, enter symbol "?" then call up required text by number

Design 1. ASU for X enterprise
Designer: Petrov, P.P.
Current assignment: "Correlate (edit) Part I of ASU concept model"
Execution period: 10 Apr 82

Frame 14. 8 Apr 82 14:10 In order to select heading, enter its number	ASU concent model Heading--Level I: 1. General description 2. Functional description 3. Structural description 4. Description of functioning 5. Appendices
Frame 15. 8 Apr 82 14:15 Edit presented text with system's editing facilities. Upon completion, enter symbol "§"	General Structure Organizationally, the ASU consists of an information system and functional subsystems based on it; technologically, it consists of the following recommended set of subsystems: material and technical equipment and market; technical preparation for production, planning for production
Frame 16. 8 Apr 82 18:00	System Reports: User: Petrov, P.P. Operating time: 3 h 50 min Degree of completion of our work: 20 % Correspondence to plan: no deviations Session ended

The user's work in the SAPR ASU begins with his being connected to the system after entering his code number (Frame 1). Let us assume that a customer who needs to introduce an industrial enterprise ASU is working with the system. He fills in the first questionnaire for rapid analysis of his enterprise that appears on the screen (Frame 2). Then, it is implied, the SAPR ASU makes the necessary primary calculations and gives the customer approximate data on the change in the characteristics of his enterprise that the introduction of an ASU will cause (Frame 3). If the customer agrees to continue the work, he is presented with a standard, maximally aggregated--but still a working--document that is a standard design concept for the corresponding type of enterprise. This document, which is oriented on the customer, gives the general characteristics of a standard model system, its capabilities and the conditions for realizing it, which serves as a basis for joint work between the customer and the ASU developer. After becoming familiarized with it, the customer formulate an agreement for correlation of the standard design concept with his enterprise and then answer the question of concluding an agreement for the designing of the ASU with the SAPR organization.

Let us assume that an agreement has been signed and the SAPR director has assigned a project leader (RP). Let us discuss the RP's work on the conceptual level; that is, when preparing an ASU design concept oriented on a specific enterprise. The RP's work consists basically of two stages: planning and monitoring the development. After being connected to the SAPR by using his own code number (Frame 4), the system for carrying out development work by scenario proposes to the project leader that he plan the appropriate work (Frame 5). In connection with this, the RP is given the capability of selecting the working mode that satisfies him best from the viewpoint of psychological comfort (Frame 6), as well as the working and auxiliary materials

that are available (Frame 7). Then the RP draw up a network chart for the appropriate work (or correct a standard one) and also indicate the actuating units by filling in a table presented to him on the screen (Frame 8). It is assumed that the RP's planning work itself is subject to monitoring by the SAPR. For instance, the system can issue warning reports about the approach of periods for the completion of work on compiling the network chart (Frame 9), in addition to monitoring the correctness of the formulation of this chart. If systems monitoring is undergone successfully, assignments are sent to the indicated actuating units and the RP moves on to the development monitoring stage.

In this stage the RP carries out current monitoring of the development process: he he reviews summaries of the state of affairs or different parts of the project, approves or returns for further development the parts of the project presented to him by the actuating units (Frames 10, 11) and, if necessary, corrects the network chart. In particular, he receives system reports on the monitoring of working periods and configurational monitoring, which involves testing different parts of the design for compatibility (Frame 12).

Let us examine the designer-editor's working scenarios. After selecting the mode for communication with the SAPR that is most convenient for him, the designer-editor requests (Frame 13) the information he needs from the design information bank (Frames 14, 15) and corrects the text of the standard design concept that is presented to him. The data needed for correlation with the object were prepared beforehand by systems analysts on the basis of a study of the object and the information supplied by the client. The designer also receives system reports on the monitoring of work execution periods and configurational monitoring.

For any category of user, a working session with the SAPR ends when he disengages from the system (Frame 16). In connection with this, information about the course of the fulfillment of the assignment is presented and certain statistics are gathered.

During the development of the proposed variant of automatic designing technology, a great deal of attention have been given to questions concerning man-computer interaction. In particular, the content of the display frames has been worked out in such a manner that nothing remains unclear to the user when he is working with the system and the answers required from him are limited to a maximally simple set of directives.

The use of the proposed method of interactive interaction, with regulation of the user's work on the basis of a scenario, contributes to clarity in the work and improves executory discipline and increases the rhythmic nature and intensity of the development work; in the final account, this results in a substantial improvement in the effectiveness of the design process and the quality of the plans that are produced.

11746
CSO: 1863/85

AUTOMATED MANAGEMENT SYSTEMS IN SOVIET INDUSTRY

Kishinev KOMMUNIST MOLDAVII in Russian No 2, Feb 83 pp 65-68

[Article by P. Soltan, Corresponding Member of the Academy of Sciences of the Moldavian Soviet Socialist Republic, and S. Kornya, Candidate in Technical Sciences: "An Important Means of Increasing Management Quality"]

[Text] It is well known that the acceleration of scientific and technical progress and the continuous increase in the scale of social production increase the volume of technical-economic information which must be processed in order for management decisions to be made. This process requires the performance of millions of computational operations, which increases the over-all volume of information processing and thus makes necessary an increase in the number of management personnel.

Time limitations and the complexity of management tasks give rise to difficulties in the realization of the possibilities of modern high-mechanized and dynamic production. Scientific and technical progress has provided the means for solving these problems: mathematical methods and computer technology for the processing and storing of information.

The practical application of economic-mathematical methods and computer technology and research performed in these areas have permitted the development of automated management systems (AMS). An AMS is the organization of the management of social production and industrial activity under the conditions of the scientific-technological revolution. Economic-mathematical methods in these systems provide for the rationalization of economic-planning calculations, while computer technology makes possible the creation of the information-technological basis of management corresponding to the current level of productive forces. This is why the party and the government are devoting considerable attention to the development and introduction of automated management systems.

This is a complex and laborious process involving the solution of a large number of technical and organizational problems. In these systems computer technology contributes to the acceleration and improvement of data processing and the carrying out of economic-planning tasks, above all tasks aimed at optimization. Since its use requires special conditions it is being introduced primarily in order to increase the management efficiency of units of the national-economic complex.

The goal of scientific management consists in directing a plant, association, branch or the entire economy of a region or of the nation as a whole such that the tasks of socialist society are performed optimally from the point of view of the fulfillment of its requirements. That is, management is a goal-directed process, organizing the productive-economic activity of appropriate units and taking into account the goals and criteria established by society, as well as the concrete possibilities of their realization.

At first glance the construction of expensive AMS appears to be a direct diversion of resources from the productive sphere. Among economic leaders the opinion is still prevalent that expenditures for these purposes are inappropriate and that the diversion of significant resources from the sphere of material production is inappropriate. The introduction of AMS, however, is not an end in itself, but rather an important means to increase management quality and on this basis to improve the productive activity of various elements of the national economy. Investment in AMS is not a diversion of resources, but a mobilization of resources to make social production more efficient.

It should be noted that there is a significant gap between the potential of the productive-economic activity of associations and plants and the results actually achieved by them. This gap creates reserves of production which are not being used by the present system of management. The magnitude of these reserves is so great that their mobilization with the aid of AMS will rapidly recoup AMS' cost.

One of the most important sources of the efficiency of automated management systems is the increase in productive output at current levels of productive capacity through better use of this capacity. In this case economic-mathematical methods and computer technology emerge as a unique capital-conserving factor; low expenditure to create AMS permits economizing on the large capital investments which would be required to develop productive capacity.

In our republic work on the creation of automated management systems at various levels has been done for the last ten years. They have been put into operation most intensively during the tenth five-year plan. In the majority of ministries and agencies under Republican or Union-Republican authority AMS are beginning to be introduced or developed. These systems are for the most part in the economic-organizational category. These are the interbranch automated systems of the State Plan (Gosplan) of the MSSR, the Central Statistical Board (TsSU) of the MSSR, the Ministry of Finance and other agencies, branch automated systems, and automated plant management systems.

All of them are being developed in accordance with the individual principle. This factor has both positive and negative features. On the one hand, this principle permits each system to develop its own strengths and do so within a short time. After all, there are as yet no planning organizations in the Republic capable of filling the orders in question. On the other hand, in practice the individual AMS are not interconnected.

Nevertheless the work which has been done is giving positive results.

It is well known that capital investment for the creation of automated management sytems and computer technology is higher than for other areas of productive-economic activity. In fact, of each ruble of overall capital investment in the country 40 kopeks go for economizing. In our Republic during the tenth five-year plan the use of AMS and computers yielded a savings of about 13.7 million rubles, and the break-even period was 4.7 years.

At the present time, in addition to the creation and introduction of local AMS, work is being done to unify them in the Republican Automated Management System (RAMS, Russian: RASU) of the national economy of the MSSR, which is one element of a nation-wide AMS. The Republican AMS is an integrated management system operating in the regime of the joint functioning of automated management systems of Republican organs (Gosplan, Gossnaba, TsSU, etc.), ministries, agencies, plants and organizations under Republican authority, as well as plants and organizations under Union authority located on Moldavian territory and interacting on the basis of the network of computer centers of the Republican system of data banks and data transmission.

Thus, the creation of the Republican AMS is a new stage in the improvement of the management of the national economy. This system possesses an entire series of new qualitative features not possessed by any of the local AMS of which it is composed. A number of problems arise in this connection which require a systemic approach.

The creation of a Republican system of data banks requires special attention. The problem here is that the documentary base of data-processing systems which is widely used at present is little suited to guaranteeing interaction between individual AMS. In the Republic there is already experience in the creation of integrated data-processing systems using the conception of data bases: at the Scientific Research Institute for Gosplan Planning of the MSSR, at the "Soyuzpishcheprom" special AMS design office, the Kishinev AMS Project-Design Office, the computer center of the Ministry of Automobile Transportation of the MSSR, and in other organizations. Practice confirms the high effectiveness of this approach.

But this is only the beginning of intense activity in the area in question. It is now necessary to create the program facilities to link up separate data banks so as to be able to shift to a Republic-wide system. All scientific research and design organizations concerned with these questions must work more intensively and expeditiously, since the development and introduction of the Republican AMS depends on them to a decisive degree.

The technical base of the RAMS is the Republican network of computer centers. Collectively utilized computer centers should constitute its core. Work has begun in the Republic to create these centers, but is not proceeding at the necessary speed. The reason for this is the absence of the required financing and an organization responsible for this activity.

Certain ministries and agencies are not showing interest, prolonging individual computer instrumentation.

Meanwhile this approach leads to excessive expenditures, and individual ministries and agencies, as a rule, are not able to acquire a powerful computer system. Yet just such machines, with a developed periphery, are required for work with data banks. This the pooling of resources to create a collective-utilization computer center is much more profitable and expedient. The creation of a network of computer centers is inseparately associated with the creation of a Republic-wide system of data transmission. Unfortunately, practically no work is being done in this area.

Solving this complex of problems will require considerable time--approximately two five-year plans. However, intensive work is required now, in the eleventh five-year plan, primarily within the framework of the Republican AMS. The high efficacy of these efforts is supported by the figures. For example, if the AMS is developed according to the individual principle during the eleventh five-year plan in the Republic, then for the period 1976-1985 savings from its introduction will comprise approximately 42.3 million rubles, and the break-even period will be 3.33 years. If, however, the AMS is created on the basis of a collective-utilization computer center, then for this period savings of approximately 57.9 million rubles may be achieved, and the break-even period will be 2.63 years. Consequently, the national economy of the Republic could obtain an additional saving of 15.6 million rubles, and the break-even period would be decreased by 0.7 years. This will require the combined efforts of all of the organizations involved, as well as scientific-research and design institutions.

The tempo of the creation and introduction of AMS and computer technology in the Republic is affected by the low organizational activity of workers at all management levels concerned with this problem. More than five years have passed since the naming of the Head Organization, the Head Supervisor and Head Designer for the development of the Republican AMS, but the posts governing their activity, as well as work on the construction of the AMS, were initiated only last year.

There was a delay in the creation of the council of the main designers through which the Head Supervisor and Head Designer of the Republican AMS oversee the scientific-research and experimental-design work.

There are also specific questions which should be resolved by various working groups: methodological questions of mathematical, programming, informational and legal maintenance, as well as questions of interbranch and branch management complexes, and others.

Training of cadres both as specialists and as economic workers, the users of the AMS, has been poor. The biggest deficit in the AMS is programmers, but no one is training them. Applied mathematicians trained by Kishinev State University are being used in this area. Another important sector is thereby stripped of workers: mathematical modeling and maintenance. The

level of training of applied economists graduated by the Republic's institutions of higher education is also low, and the task of their remedial training after graduation falls on the scientific-research institutes, design agencies and industrial facilities where they are to work. The retraining of specialists in the national economy for work in the new conditions of the automation of management processes is also weak.

We have touched on only a few key problems of great importance in the creation and introduction of the RAMS. The need for their rapid solution was indicated in the resolution of the Central Committee of the CPSU and the Soviet of Ministers of the USSR of July 12, 1979: "Improving Planning and Strengthening the Action of the Economic Mechanism on Increasing Production Efficiency and Work quality", in the documents of the 26th party congress and the November (1982) Plenum of the Central Committee of the CPSU. This will facilitate bringing to life one of the fundamental tasks posed by the party: to raise the level of planning and management, to bring them into accord with the requirements of the current stage of developed socialism.

12344
CSO: 1863/150

ACCOUNTING ISSUES BETWEEN COMPUTER EXPERTS AND USERS IN TAJIKISTAN

Dushanbe KOMMUNIST TADZHIKISTANA in Russian 9 Apr 83 p 2

[Article by V. Erokhin: "Overcoming Inertia; Policy-Technology Progress"]

[Text] In a very few minutes, the computer performs computations which required whole planning departments to spend days and sometimes even months. Who nowadays would be surprised by such a startling event? We know this, we have read about this. For many planning organizations of the republic, this is a daily routine. Two years have passed since the Gosstroy (State Committee for Construction Affairs) of TaSSR developed and introduced a complex program for automation of planning. This period is quite sufficient to judge the utility and efficiency of applying computer technology.

Recently, - recounts B. V. Sukharevskiy, Chief of the Department of Mechanization of Engineering Calculations (MIR) of the Planning Institute "Tadzh GIINTIZ" - we received a plan for a water supply system for the Kolkhoz "Kommunizm" in the Communist district.

The diameters of the water pipes which were assigned by the planners seem to have been overestimated. By changing the parameters and solving the inverse problem (all of which was done in minutes by machine) we obtained, using the computer, a more economical version of the design. A substantial amount of metal was saved, even in the planning stages. The overall economic effect approached 43,000 rubles. This is approximately 20 percent of its initial cost. The customer's bill for calculations was only 160 rubles.

In 1981, the MIR Department of the Institute became the leading organization for establishing automated planning systems on construction sites (SAPR's). Its computer center executes calculations for six planning institutes incorporated into a complex program. In addition, almost all the planning organizations of the republic have recently become customers of this computer center (VTs). A branch of the VTs of Gosstroy of TaSSR has been established in Leninabad as well.

Two years ago, plans devised using computer technology constituted just a small fraction of the total number. There were hardly any planners working

with computers. Today there are more than thirty programming resources in the bank of algorithms and programs at the VTs which serve planning organizations and which meet their requirements in the majority of divisions of planning. Up to and including the year 1982, automated calculations have been performed for 213 projects. The estimated cost of planned sites was reduced by more than 400,000 rubles. And the total annual economic effect was 351,000 rubles.

When the complex program was created, a harmonious development of the basic components - the collective-use computer centers and the centers and the services of SAPR of the planning institutes was envisioned. Let us be frank, from the beginning relations between the subdivisions did not develop in a most favorable fashion. The minutes of the board of the Gosstroy of the republic from November 24, 1982 stated: "The main handicap for wider utilization of computers in construction planning is still the lack of necessary contact and understanding between the wide circle of planners and experts dealing with the automation of planning. The services of SAPR are not being set up properly, their structure and staffing and so on are not well defined".

Accounting between the customers - the planning institutes, and the contractor - the VTs of the Gosstroy of the republic became the stumbling block.

-It may sound paradoxical, but often it does not pay for us to utilize the services of the VTs - claims V. I. Biryukov, Chief Engineer of the institute "Tadzhikgiprostroy". -I shall clarify this using one of many examples. Recently, we contracted calculations for the project "Foundations of a VTs of Stroybank of TaSSR". The Mir Department performed a quality job. Our relations, however, were complicated by a brief remark which accompanied the calculations "...the profile does not pass along the cross (lateral) force..." Thus we have incorrectly assigned the initial parameters of the elements at some point and it is now necessary to contract new calculations. It is quite expensive. We will have to pay double for the computing services -

-But how can it be otherwise - B. P. Sukharevskiy gestures helplessly. -Why should the search for an optimal solution be imposed on the budget of the VTs? These are nonproductive expenses and we have our own plan calculated in rubles -.

Both sides note the obvious imperfections in rates for automated calculations.

The tendency among the planners to return to a manual working method is beginning to crop up. On the other hand, the managers of the VTs, in order to fullfil their plan, were compelled to develop and solve auxiliary problems of the ASU [Automated Control System] for the planning institutes "Tadzhik GIINTIZ" and "Tadzhik Kolkhozproyekt". This caused the index of automation of planning in 1982 in the subdivisions of the Gosstroy to fall below the average for the republic - 2.2 percent.

In the statutes of the complex program and in the orders from Gosstroy instructions are given to the department of examination of plans not to

hand out conclusions concerning those projects which were conducted manually if they contain programming provisions. Alas, the department has not yet returned a single plan which was manually prepared.

-You must agree that it does not make sense, to say the least, to refuse a ready-made job - explains L. I. Feshchenko, Chief of the Department of examination of projects and estimates at Gosstroy.

It is difficult to agree with him, because in essence, the working discipline is being breached. One, however, can empathize with the staff of the department. Suppose they decide to follow instructions. Would the planners then immediately return to the VTs? It is doubtful whether mutual understanding between the sub-divisions can be achieved merely by using administrative measures.

-In our opinion, there is just one solution - says the chief expert of the technical division of the institute "Tadzhikgiprostvoy", L. I. Gimel'stein. -And this is leasing machine time by planners in computer centers. This has been done in other cities of the country. The SAPR groups themselves, are capable of working with computers and can code the initial parameters; they were actually taught to do this in the courses of the Mir Department. We know better how and when optimization of calculations should be carried out.

In such a case the controversy of mutual accounting will no longer arise. The utilization of computer time may serve as a clear-cut criterion for development of the basic planning index of a VTs' performance. This computer time should be compensated by the planners.

The question arises: what will the staff of the VTs do? Under these circumstances the problem of a reduction in valuable personnel may possibly arise.

-There are no contradictions here - explains L. I. Gimel'stein. -In the complex program, the functions of each one of the subdivisions are clearly defined. The immediate duties of the VTs are acquisition, mastery and supplying the planners with programming devices, introduction of a bank of algorithms and programs, training highly qualified experts of the SAPR groups and providing assistance to them.

And what is more, lately the necessity has arisen more often to establish a methodological center which would be able to perform the role of an organizer of systems of automated planning. Until now, for example, nobody actually knew how the services of the SAPR should function, only what they can do is known. Planners have at present no one to approach for assistance and they perform using the trial and error method. In the cities of Ukraine and Belorussia, the scientific-methodological investigations are assigned to the appropriate NII [Scientific-Research Institutes]. In our republic, the role of such a center (due to the absence of a specialized NII which is available in many other cities of the country) can very well be performed by the MIR Department.

It is possible that this department could be aided by experts from other republics and one should indeed request the benefit of their experience. However, for the present, such a specialized NII does not exist.

Nevertheless, a solution of all these organizational problems does not as yet fully guarantee the fullfilment of the complex program. A high increase in the level of automation of planning in the years 1984-1985 presupposes qualitative advances as well in the development of the material-technological base of Gosstroy subdivisions in the republic and planning institutes belonging to various agencies.

The possibility of establishing automated sites of planners-builders (ARMS) and user-stations in various institutes depends on whether the leaders of various agencies can find a common language. The Commission for Construction of TaSSR will have to consolidate all the participants of the complex program and this will require renovation of the existing building of the Gosstroy VTs in order to install in it yet one more third generation computer and to implement the plan of creating a unified Gosstroy center.

Towards the end of the 11th five-year plan, it is envisioned to raise the level of automation of planning in the republic to 15.8 percent. This is an impressive figure. Clearly, the road for the triumphant march of computers into the planning institutes is not paved with roses. Who knows what obstacles lie ahead? However, not only the planners and staff of the VTs, but the workers of Gosstroy and Gosplan of the republic as well, should give some thought as to how to overcome the current difficulties.

9966
CSO: 1863/133

NETWORK MEASUREMENT IN ASGS

Moscow VESTNIK STATISTIKI in Russian No 4, Apr 83 pp 41-47

[Article by N. Ivanov, deputy chief, Main Administration of Computer Work, TsSU SSSR]

[Excerpts] The Automated System for State Statistics (ASGS) has at its disposal a powerful, territorially distributed network of computer centers that function on the union, republic, oblast and rayon levels of the TsSU SSSR [USSR Central Statistical Administration] system. The basic assignments for this network are the gathering and processing of statistical and accounting information for use in the solution of planning and control problems, using a coordinated list of indicators, within the required periods of time.

At the present time, the basic computers in the computer centers are of the YeS [Unified System] type, which have increased the computer system's capabilities considerably. With the increase in capacity there has also been an increase in the volume of statistical work being done on the computers, the share of which was 17 percent of the total volume in 1975, 29.1 percent in 1980 and 32.3 percent for the first 9 months of 1982. On the union level, more than 60 percent of all the statistical information collected and processed state statistics agencies is processed on computers, and on the republic and oblast levels it has reached 40-50 percent.

Along with the growth in computer capacities and the corresponding increase in the volume of statistical information processed on computers, there appeared new technical and technological capabilities that make it possible to construct ASGS equipment and program facilities not on the basis of a set of individual (although well-equipped) computer centers, but on the basis of a unified computer network. This will facilitate the solution of problems according to a unified methodological processing program and on a unified organizational basis.

The first prerequisites for the formulation of such a network in the TsSU SSSR appeared after the creation and introduction into industrial use of the automated data bank (ABD ASGS ISKhOD) and the statistical information teleprocessing system (STOSI).

The practical experience in the use of the equipment and program facilities, together with the electronic information processing complexes that were created for the solution of specific statistical problems, demonstrated the need for and possibility of a qualitatively new approach to the solution of the question of the rational utilization of the existing computational, program and information resources of the YeS computers in the TsSU SSSR system's computer centers.

The existing methods for statistical observation of the functioning and control of the computer system are based on the collection and processing of the appropriate statistical reports and serve as a useful tool for macroanalysis of the work done by TsSU USSR's system's computer centers and stations. The data from this analysis make it possible to evaluate the level of development of the computer network as a whole by statistical observation of the work done by each computer center, primarily from the viewpoint of the utilization of computer technology facilities. At this time these data are inadequate for evaluating the reserves and losses in the use of the YeS computer facilities and working out measures for the improvement of the organizational and technical forms of their use in the computer centers and shared-use computer centers in the TsSU SSSR system. The basic shortcoming is that they are designed for utilization by a rather small circle of users. At the present time specialists from TsSU SSSR's VGPTI [All-Union State Design and Technology Institute], GUVR [Main Administration of Computer Work], NII [Scientific Research Institute], GVTs [Main Computer Center] and computer centers are working on the problem of automating the processes for calculating, planning and controlling the operation of the computer network. In connection with this, the solution of a complex of reticulometric problems is regarded as the most important task for the development of the computer system.

Reticulometry is a new field in computer network methodology that involves determining the ways and means for observing the functioning of the system and the processing and analysis of report data. The practical realization of reticulometry problems will make it possible to advance to the more efficient development of computer network architecture, with particular emphasis on its supporting subsystems--hardware, software and organizational and informational support--and the optimization of their interrelationships.

Reticulometry is based on the maximum automation of the processes of collecting and processing data with the help of a computer's own programmed measuring facilities, which are a complex of programs for the collection, processing and output of data on the use of computer technology facilities and data transmission. They should make it possible to evaluate the performance of not only a single computer, but also that of an entire set of computers under conditions of their joint operation in a computer network.

The practical realization of the complex of reticulometric problems was begun during the creation of the first stage of STOSI under the guidance of TsSU SSSR's GUVR and NII. A plan for an automated system for accounting for the use of resources has been created at the Estonian SSR TsSU's Republic Shared-Use Computer Center and introduced into practice at eight republic- and oblast-level centers in TsSU SSSR's system. The data gathered on magnetic tape at these centers were sent to the Estonian SSR TsSU's Republic Shared-Use Computer Center, where the programmed measuring facilities were used to process them on a centralized basis, after which the results were sent to TsSU SSSR's GUVR and NII. They were first used for practical purposes in 1982, during the development of standardized planning decisions for the development of ASGS's technical base during the 11th Five-Year Plan.

Let us discuss in more detail the data on the operation of YeS computers, under the control of the YeS OS [operating system], at the eight computer centers of TsSU SSSR's computer network: the Republic Shared-Use Computer Centers of the Estonian and Belorussian SSR TsSU's, the Tomskaya Oblast Statistical Administration's Shared-Use Computer Center, the Tul'skaya Oblast State Statistical Information and Computer

Center, the Republic Computer Centers of the Ukrainian, Azerbaijan and Kirgiz SSR's and the computer center of the Leningrad and Leningrad Oblast Statistical Administration. Their work was observed from the fourth quarter of 1981 through the second quarter of 1982. During this period the programmed measuring facilities collected and processed a large amount of information that characterized the operation of the computers at each computer center on the basis of the following system of indicators:
monthly data on the number of jobs carried out, the time required to do them, utilization of the central processor, number of jobs carried out on an emergency basis because of the computer and operating system malfunctions, operator errors and errors in applied programs, functioning of the operating system during the month;
information on computer operation on the days of the heaviest and lightest loads, along with the typical number of completed jobs, the time required to do them, the average amount of time required for the completion of a single job (broken down into the amount of time needed to put it into the input sequence, how long it was there, how long it was under the control of the YeS OS, how long it took to obtain the results, how much main memory was used and other data). A comparison of these data for individual computers in a single center, as well as computers in different centers, makes it possible to evaluate objectively how the YeS computer facilities are being used in the TsSU SSSR system's computer centers and shared-use computer centers. Let us concentrate on the analysis of the basic information on the operation of the YeS-1033 computer during this period that was gathered by the reticulometric facilities.

The average intensity of the flow of jobs into the shared-use computer centers was 13.3 per hour, whereas for the other TsSU SSSR computer centers that were investigated it was 12.4; the average duration of a single assignment at these centers was 4.4 and 4.1 minutes, respectively. The maximum number of jobs carried out by a single computer at the Belorussian SSR TsSU's Republic Shared-Use Computer Center in May 1982 was 3,478, and the total time required for their completion was 228.2 hours/month in the OS environment. The greatest amount of YeS OS operation time was registered at the Tomskaya Oblast Statistical Administration's Shared-Use Computer Center, in May when one computer was used for 311.8 hours to carry out 2,618 jobs. Computer centers lag behind shared-use computer centers in both time of computer operation under control of the YeS OS and number of jobs carried out.

It should be mentioned here that the quality of the job execution was quite high at all the investigated computer centers and shared-use computer centers. On the average during the observation period, the proportion of normally completed jobs carried out on the computers was 80 percent when there were at least 1,000 of them. In most cases, the amount of computer operation time lost because of emergency completion of jobs, for all causes, was about 20 percent. During the entire period of the investigation the values of these indicators were quite stable.

An analysis of the reasons for emergency completion of jobs showed that no more than 1 percent of the total computer time lost because of emergency situations can be attributed to errors in the YeS OS's operation, whereas for malfunctions in computer equipment during the solution of problems the figure was 7 percent. In the latter case 40 basic causes of emergency completion codes were isolated, among which peripheral gear malfunctions were the most important. In generalizing these data, it is possible to say that during computer operation, 8 percent of the emergency completion codes are the result of hardware malfunctions and operating system errors, whereas the other 92 percent are attributable to errors by computer operators and in the programs directly involved in processing the information.

Of these 92 percent of the emergency jobs, about 38 percent are attributable to operator error. This indicator is made up primarily of five codes that are related to infringements on the agreements provided for the computer operators by the YeS OS. The values of this indicator are not the same for different centers; as a rule, they are lower for shared-used computer centers than for the other computer centers that were investigated.

An analysis of the jobs carried out on computers indicates that there is an adequate reserve for improving the use of computer time. Here the basic reserve is an improvement in computer reliability, which includes improving maintenance, reducing the number of computer operator errors, and improving the quality of programming jobs. In connection with this, the elimination of the last two causes is an extremely substantial reserve for the better utilization of computers. For instance, the number of repeated jobs carried out on peak load days reaches 10-15 percent, whereas the number of jobs started more than once varies from 2 to 6, averaging 3 or 4.

The quality of the operators' work depends on their qualifications and more careful monitoring of the work. A reduction in the number of errors in programs can be achieved by improving the results of their debugging and the observance of stricter rules when accepting them for industrial use. If it were possible to reduce the losses of computer time due to these causes by only 50 percent, the computer system would receive about 2,000 hours of additional computer time, the cost of which is approximately 200,000 rubles. This alone could repay the expenditures for the proposed measures for utilizing additional computer time resources.

Another source for improving computer utilization under YeS OS conditions in the TsSU SSSR system's computer centers and shared-use computer centers is an increase in computer productivity when solving problems. Reticulometric data show that for the centers that were investigated, the process of carrying out jobs on computers is characterized by the following average indicators on peak load days. For instance, the average time required to enter a job in the input sequence is 4 percent of the average time the job is in the computer, whereas the average time the job spends in the input sequence is 3 percent of the average time the job is in the computer, the average time required for result output is 15 percent of the time the job is in the computer, and the time required for direct processing of the data and fulfillment of a job on a computer is 74 percent of the total computer operating time.

According to the reticulometric data, the average duration of a job on a YeS-1033 computer ranges from 6.5 to 10.4 minutes. On the average, each job consumes from 76 to 128 Kbyte of the computer's main memory. In connection with this, the input information volume averages 44 punched cards per assignment, since the original information is basically entered on magnetic tape in the shared-used computer center before the execution of a job begins. The output information volume averages 455 lines of computer alphanumeric printer computergrams.

On the average, for a single job up to three magnetic tape and up to seven magnetic disk storage units are used for a single unit for data input from punched cards and alphanumeric printers. In connection with this, the average number of magnetic disk storage units used per job is 4.6. The average number of jobs in a single operation carried out on a YeS-1033 computer with a main memory of at least 1 Mbyte is 4.1 with a dispersion of 2.5 (according to materials furnished by the Estonian SSR TsSU's Republic Shared-Use Computer Center).

What has been said indicates the necessity of the accelerated introduction of more powerful computers in the shared-use computer centers, the introduction of which would make it possible to reduce job execution time and, consequently, the time required to produce reports. Preliminary investigations show that with the introduction of the YeS-1045 computer, with a 4-Mbyte memory, the job execution time will be reduced by a factor of 1.5-2. In connection with this it is necessary to keep in mind that for the existing set of jobs in the computer centers (as was shown by the reticulometric data), processor loading time averages eight percent of the total job execution time, although the computer's main memory is used quite fully. The volumes of the planned and used main memory are characterized by the ratio 10:8; that is, the coefficient of utilization of the computer's main memory is 0.8.

An analysis of the data that have been presented showed the rationality of the proposed approach to evaluating the efficiency of computer operation on the basis of reticulometric facilities in addition to the existing ways and means used in the TsSU SSSR system. The practical realization of this approach demonstrated the possibility of maximum automation of the processes of collecting and processing data on computer resource use on the basis of programmed measuring facilities. The results of reticulometry can be used extensively to solve a large complex of problems, primarily in the working out of recommendations for the development of ASGS hardware and software. For instance, on the basis of work done by the NII and the Kazakh branch of TsSU SSSR's VGPTI on the modeling of computer technology and information teleprocessing facilities for computer centers and shared-use computer centers, plans are being made for an integrated investigation and the development of suggestion for improving equipment and programs at TsSU SSSR's computer centers in order to increase the traffic capacity of YeS computer facilities and, on this basis, to reduce ASGS problem solution time and, consequently, the cost of the development of statistical reports.

What has been said far from exhausts all the advantages of the widespread introduction of reticulometric methods and facilities. One important direction in this work is the integrated utilization of systems testing and reticulometry in computer centers on the union and republic levels under conditions of the constant functioning of the ABD ASGS ISKhOD and STOSI. They make it possible to move on to qualitatively new technology for the solution of problems in the integrated processing of statistical information, but require a substantial improvement in the reliability of the complex of computer and data transmission hardware, since the information, computation and program resources of the GVTs and republic computer centers can be joined to the computer network.

In order to maintain normal operation of the computer network, it has been proposed that every computer be equipped with a continuously operating module containing reticulometric and systems testing facilities that occupies about 30 Kbyte of the computer's main memory and carries out information switching. After a certain interval of time (5-10 minutes, for example) this module should send to the Control and Reticulometry Center (TsUS STOSI) information on the operation of each computer and the utilization of its resources. This would make it possible to realize operational regulation of the utilization of the computer network's resources, and on this basis to provide, when necessary, redistribution of the load on GVTs TsSU SSSR's and the republic shared-use computer centers' highly productive computers, present the results of the processing at any point in the network, and provide remote access to the ADB ASGS ISKhOD to economists from the GVTs and republic computer centers, and from TsSU SSSR and the TsSU's of the union republics.

At the same time, this would improve computer reliability, since operational testing and observation of the status of the computers in computer centers on the union and republic levels leads obligatorily to improvement of the entire mechanism for the maintenance and operation of computer and data transmission facilities. Expenditures for the creation of such a system, the basis of which will be TsUS STOSI, will not be large, but the effect of its introduction can be substantial, since an increase of only 1 hour of work load per year for the computers at the 16 computer centers, as achieved by reducing different types of down time, will increase the economy of the facilities substantially.

11746
CSO: 1863/164

MODELING AND AUTOMATION OF PROCESSES OF DESIGNING, MANUFACTURING AND OPERATING COMPLEX SYSTEMS

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA (NAUCHNO-PROIZVODSTVENNYY SBORNIK) in Russian No 4, Oct-Dec 82 p 60

[Article by I.I. Krinetskiy, doctor of technical sciences, and V.I. Garbarchuk, candidate of technical sciences]

[Text] The First Basin Interdepartmental Scientific and Technical Conference on "Modeling and Automation of the Processes of Designing, Manufacturing and Operating Complex Systems" was held in Odessa in June 1982. It was organized by the Ukrainian SSR Academy of Sciences' Institute of Modeling Problems in Power Engineering, the Black Sea Basin Board of the Scientific and Technical Society of Water Transportation, the Odessa Higher Engineering Nautical School imeni Leninskiy Komsomol, the Kiev Institute of Civil Aviation Engineers, the Ukrainian SSR Gosplan's Odessa Interbranch Center for Scientific and Technical Information and the Odessa Technology House. More than 80 scientists and specialists from Moscow, Leningrad, Kiev, Odessa, Khar'kov, Nikolayev, Volgograd and other cities took part in the conference. A total of 44 reports were given, and the following working sections were set up: "Automation of the Design Process," "Modeling Complex Systems," "Automation of Ship Systems and Processes."

A great deal of interest was evoked by the following plenary reports: "Scientific and Methodological Principles and the Present State of Automation of the Designing of Control Systems for Production Installations," given by Professor V.V. Solodovnikov, doctor of technical sciences and USSR State Prize laureate (Moscow), and "Natural Scientific Principles of the Automation of Intellectual Labor," given by Professor L.A. Kolesnikov, doctor of technical sciences (Khar'kov).

The reports of A.Ye. Marakhovskiy, V.S. Belinskiy and V.M. Leont'yev (Kiev) were devoted to the methodological and organizational problems of the creation of SAPR's [automated design system] in scientific research institutes and design offices. The report given by V.B. Pozdnyakov (Moscow) concerned the urgent problems of evaluating the effectiveness of SAPR's. V.I. Garbarchuk (Odessa) discussed the results of investigations of the mathematical modeling of the design process as an object for automation on the basis of a games-automaton approach. The subject of the reports given by Ye.S. Azepova (Moscow), V.S. Maksimova, A.K. Mulenko, N.V. Manoylo (Khar'kov), A.S. Presnyakov and I.M. Terterova (Leningrad) was the creation of various SAPR subsystems in electronics and instrument building.

A number of reports on the problems of modeling structures for the solution of complex problems of organizational control were presented by the Ukrainian SSR Academy of Sciences' Institute of Modeling Problems in Power Engineering (A.G. Dodonov, A.M. Shchetinin, S.P. Pelekhov, A.A. Kotlyarenko, V.M. Shishmarev and others). Other reports that were heard concerned questions on the operation of automated ship systems, prospects for the development of research into the use of sailing ships in the transport fleet, and the modeling and design problems connected with them.

Many scientific centers, scientific research institutions, VUZ's and design offices in this country are concerned with problems in the automation of the design process. There exists a multitude of packages of applied programs for the solution of individual design problems in electronics, shipbuilding, machine building and the building of cities. However, the plans for investigations of the automation of the design process are still insufficiently coordinated. Therefore, the main purpose of the conference was the exchange of information and the results of research on modeling and automating the design process among scientists and specialists from different scientific schools and departments. In the future, this will make it possible to develop automated design systems more efficiently (from the systems viewpoint) and solve other related problems. The exchange of opinions on the state of the problem as a whole--from the need substantiation stage the stage of operating systems that have been created--also proved useful.

The conference passed a resolution aimed at improving the coordination of research in this field.

11746
CSO: 1863/85

PROSPECTS FOR DEVELOPMENT AND USE OF MICROPROCESSORS AND MICROCOMPUTERS

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA (NAUCHNO-PROIZVODSTVENNYY SBORNIK) in Russian No 4, Oct-Dec 82 p 61

[Article by Yu.A. Ilovayskiy, candidate of technical sciences, member, Ukrainian Republic Board, Scientific and Technical Society of the Instrument Building Industry imeni Academician S.I. Vavilov]

[Text] A scientific and technical seminar on "Prospects for the Development and Use of Microprocessors and Microcomputers" was held in Kiev from 13 to 15 September 1982. It was organized by the Ukrainian Republic Board of the Scientific and Technical Society of the Instrument Building Industry imeni Academician S.I. Vavilov, and was held at the Kiev Branch of the Moscow Institute for the Improvement of Qualifications.

Specialists gave the following reports at the seminar: "On the Prospects for Introducing Microprocessor Technology (MPT) Into Instruments and Control System's at Minpribor [Ministry of Instrument Making, Automation Equipment and Control Systems] Enterprises" (Candidate of Technical Sciences, A.S. Semin, chief, Microelectronics Department, Minpribor); "Automating the Designing of MPT" (Canidate of Technical Sciences A.D. Mil'ner); "On Techniques for Training MPT Specialists in the Institute for the Improvement of Qualifications" (Candidate of Technical Sciences Yu.A. Ilovayskiy and D.A. Beznosenko); "The Kiev Branch for the Improvement of Qualifications as a Center for Scientific and Technical Propaganda About Achievements and Progressive Experiences in Using MPT" (Candidate of Technical Sciences V.I. Men'kovskiy and V.D. Bakumenko); "KTS LIUS-2 [Hardware Complex for Local Information and Control Systems, Model 2] Aggregate Complex of Hardware Based on Microprocessors for local ASU TP's [Automated System for the Control of Technological Processes" (Yu.V. Rozen); "On the Use of MPT in ASU's" (Candidate of Technical Sciences Yu.I. Artemov); "Organization of Interactive Modes for Monitoring Logic Units, Based on the 'Elektronika-60' Microcomputer" (Ye.Z. Perel'rayzen and others).

The participants in the seminar noted the feasibility of the widespread introduction of microprocessors in microcontrollers, information and control systems and local and distributed ASU TP's and the need for the development of methodological recommendations for their use. In view of the urgency of these problems, it was decided to conduct seminars on this subject on a regular basis.

11746
CSO: 1863/85

'AUTOMATION-83' EXHIBIT IN MOSCOW

Riga SOVETSKAYA LATVIYA in Russian 17 May 83 p 4

[Article by Yu. Sinyakov: "Automation -83 in Moscow"]

[Text] From the number of exhibitors, countries represented and transactions anticipated, the "Automation-83" exhibit will become one of this year's major reviews of technology. It will be held in Moscow from May 25 through June 8.

The coming demonstration of modern automation methods--"mini" and "micro" computers, industrial robots, machine tools with digital-program control, programmable monitors--is attracting the attention of specialists around the world, the Novosti Press Agency correspondent at the USSR Industrial Trade Board expocenter has been informed. Twenty-four countries and West Berlin haveeagreed to participate in the exhibition. Besides associations and enterprises from Bulgaria, Hungary, the German Democratic Republic, Poland, Rumania, Czechoslovakia and Yugoslavia, with which the Soviet Union has worked for many years in the field of the automation of industrial processes, firms from Great Britain, Italy, France, Japan and the USA will mount their exhibits in Moscow.

Our country will present interesting exhibits in its pavilion at the exhibition. Among them will be instruments and devices intended for use in loading branches of industry, health care and scientific research. Thus, the new technical methods being exhibited reflect a fundamentally new stage in the development of automated control systems in Soviet machine building. They embrace the entire process of the construction of a given mechanical component: from design to production and testing. This is a question of a kind of integration of extremely complex tasks in the control of a modern plant. This is not only a matter of forecasting and design, fundamental and auxiliary activity in production, but also a matter of marketing and supply. One of the most important parts of such a system, which will be demonstrated at the Soviet pavilion, is the automation of the operative control of production within the framework of section, team and work place, as well as technical complexes consisting of robots, digitally controlled machine tools and testing stations. It is here that the foundations of the technology of the future are being laid: to begin with so-called flexible production, and then unmanned plants, in which human participation in manufacturing operations will be reduced to a minimum.

Automation, which is penetrating all areas of our life, is becoming the object of broad international collaboration. The Soviet Union's most fruitful collaboration is with the fraternal socialist countries. The Soviet exhibits will tell of a series of joint efforts based on agreements with other nations--members of the Council on Economic Cooperation. Here, for example, it will be possible to see in operation a system for the production of polyethylene. Developed by Soviet specialists and their colleagues from the German Democratic Republic within the framework of the "Polimir" joint program, it has already been introduced into the economy and is successfully operating in the large-scale production of polyethylene in the city of Sumgait (USSR) and the "Leina-Werke" national enterprise (GDR).

The result of the joint efforts of Soviet and Bulgarian designers is an automated control system for machine-building and electronics plants, while the joint effort of Soviet and Polish engineers are directed toward the design of an automated control system for sulfuric acid production.

The automated control systems to be shown at the exhibition are used in various countries around the world. The technical components of these systems are now being supplied by the Soviet Union to a Czechoslovak tube mill in the city of Khomutov, the Jose Marti Airport in Cuba, and metallurgical combines in Bulgaria and Rumania, as well as enterprises in India, Pakistan, Turkey, Iran and Nigeria.

In discussing the exhibits the director of the Soviet section of the "Automation-83" exhibition Al'gis Gurauskas noted the following fact: all of the automated control systems being built or designed in the USSR are based on an international family of computers.

This family consists of electronic computers produced in various countries of the Council on Economic Cooperation, in accordance with a unified program of cooperation and specialization. From 1981 through 1985 joint procurement of computers of the "maxi", "mini" and "micro" classes will approximately double as compared to the preceding five-year plan.

"Technical problems of extraordinary complexity can be solved through socialist integration", said Al'gis Gurauskas in conclusion. "And there is no doubt that, at international reviews of technology of this sort, a unified family of robots produced by the CEMA member nations will participate along with the unified computer system".

12344
CSO: 1863/150

PUBLICATIONS

SYNOPSES OF ARTICLES IN UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, MAR-APR 83

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 83 pp 128, 129, 131, 133, 135

UDC 681.322

CHOICE OF DATA PROCESSING SUBSYSTEM STRUCTURE OF MULTIPROCESSOR COMPUTER SYSTEMS WITH REARRANGEABLE STRUCTURE

[Synopsis of article by Anatoliy Alekseyevich Zabolotnyy, senior scientific associate, NIIUVM (Severodonetsk), Vladislav Valentinovich Ignatushchenko, candidate of technical sciences, IPU AN SSSR (Moscow), Vladimir Mikhaylovich Kostelyanskiy, candidate of technical sciences, Galina Mikhaylovna Lekhnova, engineer, and Dmitriy Aleksandrovich Nedzel'skiy, senior scientific associate, NIIUVM (Severodonetsk), pp 3-6]

[Text] Requirements imposed upon data processing subsystems of multiprocessor computer systems with a rearrangeable structure are considered. The influence of a combination of scalar and vector instructions on the performance efficiency is investigated, and recommendations given as to the advisability of the combination.

UDC 681.325.33:519.712.5

UNIVERSALITY OF DISCRETE SIGNAL CONVERSION. GENERATION AND RECEPTION DEVICES

[Synopsis of article by Sergey Mikhaylovich Krylov, engineer, Kuybyshev Polytechnical Institute (Kuybyshev), pp 7-10]

[Text] A model is considered of a general purpose processor of conversion of a parallel code to a sequence of discrete signals and vice versa. The model permits a simple technical interpretation which underlies the development of practical variants of universal programmable processors of a lower hierarchical level.

UDC 621.3.019

A METHOD OF ESTIMATING RELIABILITY OF NONREPAIRABLE TECHNICAL SYSTEMS ON THE BASIS OF PROBABILITY-PHYSICAL MODELS OF FAILURE DISTRIBUTION

[Synopsis of article by Valeriy Pavlovich Strel'nikov, candidate of technical sciences, SKB MMS IK AN USSR (Kiev), pp 11-13]

[Text] A new method is considered of estimating the reliability of nonrepairable technical systems. It uses probability-physical (parametric) models of reliability. The accepted, generalized defining parameter of the system makes it possible to change from the multivariate distribution of probabilities of smooth running of an irredundant system to the univariate one.

UDC 681.325.65

SIMULATION METHOD OF ESTIMATING THE CORRECTION EFFICIENCY BY MEANS OF HARDWARE CHECK OF ERRORS IN DISCRETE COMPUTER OBJECTS

[Synopsis of article by Boris Yefimovich Liokumovich, senior scientific associate, NIIEVM (Minsk), pp 14-17]

[Text] A solution to the problem of estimating the efficiency of error correction by hardware check and determining a structure of errors in discrete computer objects is considered based on computer simulation of the object operation and the check means. Results are given of investigation into the influence of malfunctions and failures of elements of the medium level of integration on the error structure, and the probability of error detection by modulo check circuits.

UDC 681.3.06:62-507

APPLICATION OF RESIDENT INTERPRETERS METHOD TO MICROPROCESSOR REALIZATIONS OF LOGICAL CONTROL ALGORITHMS

[Synopsis of article by Gennadiy Ivanovich Ivanov, candidate of technical sciences, TRTI (Taganrog), pp 18-21]

[Text] The application is described of the method of resident interpreters to implementation of a virtual special-purpose logical controller on the basis of the K580IK80 general-purpose microprocessor. The controller is designed to execute logical control algorithms formulated in a modified language of algorithm logical schemes.

UDC 681.325.5-181.4

CONSTRUCTION OF DIGITAL DEVICES WITH MICROPROCESSORS AND PROGRAMMABLE LOGICAL MATRICES

[Synopsis of article by Valeriy Anatol'yevich Sklyarov, candidate of technical sciences, Minsk Radiotechnical Institute (Minsk), pp 21-23]

[Text] A digital system consisting of a control device which interacts with a controlled object is considered. A structure of the device is suggested, based on a microprocessor set and a control automaton on programmable logical matrices.

UDC 681.3.51./6.42

AN APPROACH TO THE PROBLEM OF ASSEMBLING TECHNICAL SYSTEMS

[Synopsis of article by Guriy Mikhaylovich Bogdanov, candidate of technical sciences, Novgorod Polytechnical Institute (Novgorod), and Aleksandr Ivanovich Polovinkin, doctor of technical sciences, Ivanov Energy Institute (Ivanovo), pp 24-27]

[Text] A wide range of problems of assembling complex technical systems from functional units which can occupy different positions relative to eabh other in space are considered. Quantitative indices of functional units assembly interaction and a criterion of assembly perfection of the technical system are suggested. An assembly problem solution algorithm is described.

UDC 62.50:007

A METHOD OF SIMULTANEOUS SOLUTION OF PROBLEMS OF COMPONENT LAYOUT AND CONNECTIONS ROUTING

[Synopsis of article by Yuriy Frantsevich Zin'kovskiy, doctor of technical sciences, Anatoliy Nikolayevich Usatenko, engineer, and Valentin Vladimirovich Parakhin, engineer, KPI (Kiev), pp 27-32]

[Text] A method of simultaneous solution of component layout and connections routing problems is presented. Principles are considered of construction and control of a self-adjusting subsystem of computer-assisted design of microelectronic equipment units.

UDC 621.3.049.75.001.63:681.3.06

A DIALOG SYSTEM OF DESIGNING PRINTED CIRCUIT BOARDS AT AN AUTOMATED WORK PLACE COMPLEX

[Synopsis of article by Aleksandr Yakovlevich Vol'fenson, engineer (Gor'ky), Gennadiy Pavlovich Demidov, engineer, "Elektromash" plant (Gor'ky), David Yefimovich Zapolotskiy, engineer, "Elektromash" plant (Gor'ky), Vladimir Ivanovich Peskov, engineer (Gor'ky), and Davyd Mikhaylovich Shteyman, engineer, Scientific Research Institute of Mechanics at Gor'ky State University (Gor'ky), pp 32-36]

[Text] A dialog system for two-sided printed circuit board design automation is described. Requirements taken into account in the process of system development are given. Principle characteristics of its subsystems are considered.

UDC 681.324.001.57

REDUCTION OF TIME OF COMPUTER SYSTEM SIMULATION

[Synopsis of article by Vladimir Dmitriyevich Yefremov, doctor of technical sciences, Vladimir Anatol'yevich Zhvarikov, candidate of technical sciences, and Yuriy Alekseyevich Kurochkin, engineer, LPI (Leningrad), pp 36-41]

[Text] A method for reducing the time of computer system simulation by reducing the simulated operating range of computer systems is proposed. While retaining the performance of the units comprising the computer system, the volumes of problems being solved are recalculated according to the new simulation interval.

UDC 681.3.06.2

SOFTWARE OF A DATA TELEPROCESSING SYSTEM RELYING ON DATA RECORDERS

[Synopsis of article by Vladimir Sergeyevich Bespalov, engineer, NPO "Lenelektron-mash" (Leningrad), pp 42-44]

[Text] Based on studies on the development and implementation of a data teleprocessing system software, an application program package is offered to control data recorders by means of a remote communication device in rapid data collection systems in automated control systems using communication channels. Questions related to the package functions are considered.

UDC 681.327.2:621.3.019.3

MATHEMATICAL MODELS OF INDICES OF ACTIVITY OF A KEYBOARD DEVICE OPERATOR

[Synopsis of article by Rafail Samuilovich Grayfer, engineer, RVTs TsSU LitSSR (Vil'nius), pp 44-48]

[Text] A problem of estimating the error-free activity of an operator of keyboard manual data input devices is examined. A classification of structure types of these devices is given. Structures of activity of the operator of the manual input device are described, and the corresponding mathematical models suggested. Analytical models are checked with data of two independent experiments.

UDC 519.685.4

SUPPORT OF FUNCTIONS OF THE OPERATING SYSTEM OF THE "EL'BRUS" MULTIPROCESSOR COMPUTER COMPLEX IN THE TEMP INSTRUMENTAL COMPLEX

[Synopsis of article by Aleksandr Abramovich Gutman, engineer, NF ITM and VT AN SSSR (Novosibirsk), pp 49-52]

[Text] A problem is considered of support of functions of an operating system and of another software of a goal-oriented computer in instrumental complexes. Two solutions to this problem, accepted in the TEMP instrumental complex which simulates the "El'brus" multiprocessor computer complex on the BESM-6, are given.

UDC 681.3.51.16.42

ORGANIZATION OF THE PROCESS OF PROBLEM-SOLVING PLANNING IN OPERATING SYSTEMS DESIGNED ON THE HIERARCHICAL PRINCIPLE

[Synopsis of article by Yevgeniy Kirillovich Kurako, candidate of technical sciences (Moscow), pp 53-56]

[Text] A structure organization of an operating system designed on the hierarchical principle for a general-purpose computer is considered. A principle is described for constructing the process of problem-solving planning for an operating system with four levels of computation.

UDC 681.3.06

SOFTWARE DEBUGGING IN A DIALOG MULTITERMINAL SYSTEM

[Synopsis of article by Yuriy Dmitriyevich Kolyakin, engineer, GPI "Proyektavtomatika", (Magnitogorsk), pp 56-60]

[Text] Principle characteristic features and a structure of a debugging system implemented in a dialog multiterminal system are considered. A structure of the data base filled during test runs and containing, in compact form, a trace of events in programs being tested is described. Examples are given of using language facilities.

UDC 681.322.012.066

REORGANIZATION OF A SINGLE-PROCESSOR REAL-TIME OPERATING SYSTEM INTO A MULTIPROCESSOR ONE

[Synopsis of article by Artur Lendrushovich Agaronyan, junior scientific associate, and Mikhail Iosifovich Belyakov, candidate of technical sciences, INEUM (Moscow), pp 60-65]

[Text] Construction is described of a multiprocessor complex operating system using the SM computer. An interprocessor communication adapter is used as an interface. Transformation of a single-processor real-time operating system into one that controls a complex is examined. Fragments of algorithms are given.

UDC 681.322

A METHOD OF HORIZONTAL FORECASTING AND CONNECTED STACKS IN SYNTACTICAL ANALYSIS

[Synopsis of article by Nikolay Koronadovich Bayushev, engineer, and Sergey Nikolayevich Kazakov, engineer, "Elektromash" plant (Gor'kiy), pp 65-68]

[Text] A method of syntactical analysis for constructing high-level translators is suggested which is oriented toward structures stipulated by a CF-grammar and provides a comparatively simple and efficient algorithm of syntactical analysis. The method is described based on experience in practical realization of the programming system.

UDC 681.3.06.44

ONE APPROACH TO ESTIMATING THE REQUIRED LEVEL OF VALIDITY OF DATA PREPARATION WHEN DESIGNING ASU SOFTWARE

[Synopsis of article by Petr Mikhaylovich Baranenko, candidate of technical sciences (Petrodvorets), pp 69-71]

[Text] An approach to and methods of estimating the required level of data preparation validity when designing ASU software using programming automation facilities are suggested. An example is given of estimation for a computer center of a scientific-technical nature.

UDC 62-52:681.3.06

DATA STORAGE AND CONTROL SYSTEM

[Synopsis of article by Vladimir Makarovich Savinkov, candidate of technical sciences, VNIIPOU (Moscow), and Viktor Leont'yevich Shamis, engineer, VGPTI TsSU SSSR (Moscow), pp 72-74]

[Text] A data storage and control system is considered, designed for use in the design of systems processing accounting-statistical and other economic information. The system is performed as an application program package. Input language operators have a high information-computing capacity that significantly reduces labor costs for the automated system design.

UDC 681.3.06/94

METHOD OF DATA ORGANIZATION FOR SIMULATING VIRTUAL EXTERNAL MEMORY OF YeS COMPUTERS AND CREATING LARGE-CAPACITY DATA BASES

[Synopsis of article by Leonid Ivanovich Semchenkov, senior instructor, Novopolotsk Polytechnical Institute (Novopolotsk), pp 74-79]

[Text] A new method of data organization on external media of the YeS computer memory is considered, which makes it possible to automate allocation of a multilevel external memory and create data bases of practically unlimited capacity. Characteristics of a software of the new method are given. Application of this method in practice is illustrated by the example of the data set control system PROTON, realized in the YeS OS medium and having automatic self-organization facilities.

UDC 681.3.06

REALIZATION OF A BASE INFORMATION SYSTEM BASED ON THE OKA DATA BASE CONTROL SYSTEM

[Synopsis of article by Igor' Leonidovich Chudinov, candidate of technical sciences, and Vladimir Zakharovich Yampol'skiy, doctor of technical sciences, Tomsk Polytechnical Institute (Tomsk), pp 80-82]

[Text] The composition and principle functions are given of a general-purpose information system using the OKA data base control system. Some characteristics of its use in a branch ASU of the Ministry of Higher Education of the USSR are given.

UDC 681.3.06

APPLICATION OF THE OKA DATA BASE CONTROL SYSTEM FOR STORAGE AND PROCESSING OF HEAVILY LOADED ORIENTED GRAPHS OF LARGE DIMENSIONALITY

[Synopsis of article by Yevgeniyy Zakharovich Zinder, senior scientific associate, MIIT (Moscow), and Georgiy L'vovich Tsipes, engineer, RIVTs Minzdrava RSFSR (Moscow), pp 83-85]

[Text] An adequate method of convenient and reliable maintenance of large oriented graphs (B-graphs) as data bases of the OKA data base control system of a special design, and of their processing in various modes, is described. Time characteristics of the B-graph access and a method of estimating the time of solving special problems (including combinatorial ones) directly on the B-graph are given.

UDC 681.3.06/94

DATA BASE FOR INFORMATION-MEASUREMENT SYSTEM IN A REAL-TIME DOS

[Synopsis of article by Mikhail Georgiyevich Logunov, engineer, and Ol'ga Ivanovna Pimenova, engineer, IPU AN SSSR (Moscow), pp 86-89]

[Text] An application program package implemented in the real-time DOS for the M6000 computer is considered, which provides accumulation, storage and modification of data in disk files.

UDC 681.3.016

ORGANIZATION OF SINGLE-TRIAL SEARCH IN STATIC FILES

[Synopsis of article by Eduard Aleksandrovich Dedikov, candidate of technical sciences, Vitaliy Fedorovich Chentsov, candidate of technical sciences, KhIRE (Khar'kov), and Oleg Timofeyevich Til'chin, candidate of technical sciences, Khar'kov Engineering and Economics Institute (Khar'kov), pp 89-90]

[Text] A method of perfect addressing is suggested which provides a single-trial search through the mapping of a set of records into a set of addresses and a sufficiently high density of file filling. Results of numerical experiments on a computer which support the efficiency of this method are given.

UDC 681.3.068

CONDITIONS OF THE APPLICATION OF THE RANDOMIZATION METHOD DURING REQUEST TRANSLATION

[Synopsis of article by Nikolay Pavlovich Smolin, engineer (Omsk), pp 90-92]

[Text] An estimate is given of request translation by tables and translation using time and memory randomization. The expressions obtained enable determination of the efficiency of applying the randomization method in the process of request translation for information reference systems.

UDC 681.3.068

SPECIAL-PURPOSE ARCHIVAL SERVICE FOR SOLVING LOGICAL EQUATIONS ON YeS COMPUTERS

[Synopsis of article by Aleksandr Igorevich Gedike, engineer, Scientific Research Institute of Automatics and Electromechanics (Tomsk), pp 92-95]

[Text] The organization and functional possibilities of an archival service are described. The service is based on YeS computers and is intended for storage of systems of logical equations and their conversion at various steps of solution.

UDC 681.322.066

PARALLEL DATA EXCHANGE WITH EXTERNAL MAGNETIC DISK MEMORIES

[Synopsis of article by Vladimir Andreyevich Savenko, engineer, GlavNIIVTs Gosplana USSR (Kiev), and Vladimir Il'ich Dvortsin, candidate of technical sciences, KIIGA (Kiev), pp 95-98]

[Text] A method of organizing parallel data exchange with distributed files on magnetic disk memories switched to different I/O channels is analyzed.

UDC 658.012.011.056:681.3.06

A COMPUTER-ASSISTED KINETIC EXPERIMENT CONTROL SYSTEM

[Synopsis of article by Abdufattakh Il'khamovich Gulyamov, engineer, Khikmatulla Kudratovich Kasymov, engineer, Masuddzhan Fatkhullayevich Makhmudov, engineer, Dilyara Akhmetovna Sadykova, candidate of technical sciences, and Talikh Gulyamovich Shamsiyev, candidate of technical sciences, IK s VTs UzSSR (Tashkent), pp 99-102]

[Text] A two-level system of controlling the number of pilot technological plants for investigation of petroleum chemistry processes is considered. A structure and composition of system and application software which fits the system requirements are given.

UDC 62-501.7

STANDARD MICRO INFORMATION COMPUTING COMPLEXES AND AUTOMATED WORK PLACES FOR RESEARCHERS IN BIOLOGICAL LABORATORIES

[Synopsis of article by Aleksandr Pavlovich Kazantsev, engineer, and Lev Aleksandrovich Pronin, engineer, Institute of Photosynthesis, AN SSSR (Pushchino), pp 102-106]

[Text] Characteristic features of the automation of mass biological experiments are examined. A number of configurations of automated work places for researchers based on the "Elektronika DZ-28" desk computer and the "Elektronika-60" microcomputer are described. These micro information computing complexes and automated work places for researchers are used in biophysics and biology as well as in other branches of science and technology.

UDC 681.322

DATA COLLECTION AND ANALYSIS SYSTEMS IN MEDIUM- AND LOW-ENERGY NUCLEAR PHYSICS

[Synopsis of article by Rafail Grigor'yevich Ofengenden, doctor of technical sciences, IYaI AN USSR (Kiev), pp 106-110]

[Text] Characteristic features of data collection and analysis systems intended for research automation in the field of medium- and low-energy nuclear physics are considered. A system developed at the Institute of Nuclear Research of the Ukrainian Academy of Sciences for data measurement and processing in a unique isochronal cyclotron is briefly described.

UDC 681.3.06

GRAPHIC PACKAGE FOR DATA ANALYSIS. STRUCTURE AND MAIN PRINCIPLES

[Synopsis of article by Leonid Georgiyevich Kaminskiy, candidate of physical-mathematical sciences, Stanislav Vladimirovich Klimenko, candidate of physical-mathematical sciences, Valentin Nikolayevich Kochin, junior scientific associate, Aleksandr Vadimovich Samarin, engineer, and Aleksandr Petrovich Sokolov, junior scientific associate, Institute for High Energy Physics [IFVE], (Serpukhov), pp 111-114]

[Text] Main construction principles, functional possibilities, structure and characteristic features of a graphic package ATOM are described. The package is designed for employment in experimental physics data analysis.

UDC 681.3.06

A PACKAGE FOR DESIGNING HARDWARE-INDEPENDENT SOFTWARE OF COMPUTER GRAPHICS

[Synopsis of article by Gennadiy Aleksandrovich Pankeyev, engineer, NF ITM and VT AN SSSR (Novosibirsk), pp 115-117]

[Text] Construction principles, main capabilities add some problems of realization of a graphic subprograms dialog package ANEGRAF are described. The package makes it possible to design hardware-independent application programs. A graphic protocol is implemented that permits using this package for a computer network.

UDC 621.774.65.011.46:681.3

IMPROVEMENT OF TECHNICAL AND ECONOMIC PLANNING IN INDUSTRIAL COMPLEX BASED ON MULTIVARIANT OPTIMIZATION CALCULATIONS

[Synopsis of article by Aleksandr Mikhaylovich Vaynzof, candidate of economic sciences, Lyudmila Ivanovna Kondrashova, junior scientific associate, and Zinaida Vasil'yevna Mashkova, junior scientific associate, VNITI (Dnepropetrovsk), pp 118-120]

[Text] An automated system of multivariant calculations of the current production plan in terms of natural and cost indices at the level of industrial complexes is considered. The system makes it possible to draw up an optimal production plan and to perform multivariant calculations.

UDC 384.984

A DIALOG ALGORITHM OF MULTIPURPOSE OPTIMIZATION OF A PRODUCTION PLAN USING LINGUISTIC VARIABLES

[Synopsis of article by Mikhail Iosifovich Mel'tser, candidate of technical sciences, LIMTU (Leningrad), and Gennadiy Aleksandrovich Strugach, engineer, LETI (Leningrad), pp 120-126]

[Text] An algorithm of drawing up a production plan of marketable products is suggested which satisfies the system of specified technical and economic indices. The search for a solution is carried out through a dialog with management personnel under conditions of uncertain source information. Examples and results of the experiment are given.

9875
CSO: 1863/140

- END -